

# **MultiMediaCard Product Manual**

*This manual covers the SanDisk MultiMediaCard which was developed by SanDisk's Design Center located in Tefen, Israel. The MultiMediaCard supports version 2.11 of the MultiMediaCard Specification.*



CORPORATE HEADQUARTERS

140 Caspian Court

Sunnyvale, CA 94089-1000

408-542-0500

FAX: 408-542-0503

URL: <http://www.sandisk.com>

*SanDisk® Corporation general policy does not recommend the use of its products in life support applications where in a failure or malfunction of the product may directly threaten life or injury. Per SanDisk Terms and Conditions of Sale, the user of SanDisk products in life support applications assumes all risk of such use and indemnifies SanDisk against all damages.*

*The information in this manual is subject to change without notice.*

*SanDisk Corporation shall not be liable for technical or editorial errors or omissions contained herein; nor for incidental or consequential damages resulting from the furnishing, performance, or use of this material.*

*All parts of the SanDisk MultiMediaCard documentation are protected by copyright law and all rights are reserved. This documentation may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from SanDisk Corporation.*

*SanDisk and the SanDisk logo are registered trademarks of SanDisk Corporation.*

*Product names mentioned herein are for identification purposes only and may be trademarks and/or registered trademarks of their respective companies.*

© 2001 SanDisk Corporation. All rights reserved.

*SanDisk products are covered or licensed under one or more of the following U.S. Patent Nos. 5,070,032; 5,095,344; 5,168,465; 5,172,338; 5,198,380; 5,200,959; 5,268,318; 5,268,870; 5,272,669; 5,418,752; 5,602,987. Other U.S. and foreign patents awarded and pending.*

Lit. No. 80-13-00089 Rev 3 7/2001

Printed in U.S.A.

#### **Revision History**

- Revisions dated before 1/98 – initial release and general changes.
- Revision dated 1/98 – general editorial changes, manual reorganized, technical changes to reflect support of MultiMediaCard Specification version 1.3, new timing diagrams added. Pin 6 definition changed in SPI mode from SPI select to VSS2 (supply voltage ground).
- Revision dated 4/98 – changes reflect support of MultiMediaCard Specification version 1.4, updated timing for Multiple Write with no Busy, updated SPI command class definition, added Error Protection section, changed operating temperature specification to -25° to 85°C.
- Revision dated 4/28/98 – Updated C\_SIZE and C\_SIZE\_MULT field definitions.
- Revision 1 dated 4/99 – Added 32 MB MultiMediaCard, general technical and editorial changes, added power up section.
- Revision 2 dated 4/2000 – Changed mechanical specification drawing, clarified system performance specifications, editorial changes.
- Revision 3 dated 7/2001 – Manual reformatted, new capacities and specifications added. Changes reflect support of MultiMediaCard Specification, version 2.11.

# Table of Contents

1.0	Introduction to the MultiMediaCard.....	7
1.1	Scope .....	8
1.2	Product Models .....	8
1.3	System Features.....	8
1.3.1	MultiMediaCard Association Standard Features.....	8
1.4	MultiMediaCard Standard.....	9
1.5	Functional Description .....	9
1.5.1	Flash Technology Independence .....	10
1.5.2	Defect and Error Management.....	10
1.5.3	Endurance .....	10
1.5.4	Wear Leveling .....	11
1.5.5	Using the Erase Command.....	11
1.5.6	Automatic Sleep Mode.....	11
1.5.7	Hot Insertion.....	11
1.5.8	MultiMediaCard Mode .....	11
1.5.8.1	MultiMediaCard Standard Compliance .....	11
1.5.8.2	Negotiating Operation Conditions .....	11
1.5.8.3	Card Acquisition and Identification.....	12
1.5.8.4	Card Status.....	12
1.5.8.5	Memory Array Partitioning.....	12
1.5.8.6	Read and Write Operations .....	14
1.5.8.7	Data Protection in the Flash Card.....	15
1.5.8.8	Erase.....	15
1.5.8.9	Write Protection .....	15
1.5.8.10	Copy Bit.....	16
1.5.8.11	The CSD Register .....	16
1.5.9	SPI Mode .....	16
1.5.9.1	Negotiating Operating Conditions.....	16
1.5.9.2	Card Acquisition and Identification.....	16
1.5.9.3	Card Status.....	16
1.5.9.4	Memory Array Partitioning.....	16
1.5.9.5	Read and Write Operations .....	17
1.5.9.6	Data Transfer Rate .....	17
1.5.9.7	Data Protection in the MultiMediaCard .....	17
1.5.9.8	Erase.....	17
1.5.9.9	Write Protection .....	17
2.0	Product Specifications .....	18
2.1	System Environmental Specifications .....	18
2.2	Typical System Power Requirements.....	18
2.3	System Performance .....	19
2.4	System Reliability and Maintenance .....	19
2.5	Physical Specifications.....	19
3.0	Installation .....	21
3.1	Mounting.....	21
4.0	MultiMediaCard Interface Description.....	22
4.1	Physical Description .....	22
4.1.1	Pin Assignments in MultiMediaCard Mode.....	22

## Table of Contents (continued)

4.1.2	Pin Assignments in SPI Mode .....	22
4.2	MultiMediaCard Bus Topology .....	22
4.2.1	Power Protection .....	23
4.3	SPI Bus Topology .....	24
4.3.1	Power Protection .....	25
4.4	Electrical Interface .....	25
4.4.1	Power-up .....	25
4.4.2	Bus Operating Conditions .....	26
4.4.3	Bus Signal Levels .....	27
4.4.4	Open-Drain Mode Bus Signal Level .....	28
4.4.5	Push-pull Mode Bus Signal Level .....	28
4.4.6	Bus Timing .....	28
4.5	MultiMediaCard Registers .....	29
4.5.1	Operating Conditions Register (OCR) .....	29
4.5.2	Card Identification (CID) Register .....	31
4.5.3	CSD Register .....	31
4.5.4	Status Register .....	38
4.5.5	RCA Register .....	40
4.5.6	MultiMediaCard Registers in SPI Mode .....	40
4.6	File System Format .....	40
5.0	MultiMediaCard Protocol Description .....	43
5.1	General .....	43
5.2	Card Identification Mode .....	44
5.2.1	Reset .....	45
5.2.2	Operating Voltage Range Validation .....	45
5.2.3	Card Identification Process .....	46
5.3	Data Transfer Mode .....	47
5.3.1	Data Read Format .....	48
5.3.2	Data Write Format .....	49
5.4	Clock Control .....	51
5.5	Cyclic Redundancy Codes (CRC) .....	52
5.6	Error Conditions .....	53
5.6.1	CRC and Illegal Command .....	53
5.6.2	Read, Write and Erase Time-out Conditions .....	54
5.7	Commands .....	54
5.7.1	Command Types .....	54
5.7.2	Command Format .....	55
5.7.3	Command Classes .....	55
5.7.4	Detailed Command Description .....	56
5.8	Card State Transition Table .....	60
5.9	Responses .....	62
5.9.1	Data Response .....	63
5.10	Timings .....	63
5.10.1	Command and Response .....	63
5.10.2	Data Read .....	65

## Table of Contents (continued)

5.10.3	Data Write.....	66
5.10.4	Timing Values .....	68
6.0	SPI Mode .....	69
6.1	Introduction .....	69
6.2	SPI Interface Concept.....	69
6.3	SPI Bus Topology .....	69
6.4	MultiMediaCard Registers in SPI Mode .....	71
6.5	SPI Bus Protocol .....	71
6.5.1	Mode Selection .....	71
6.5.2	Bus Transfer Protection.....	72
6.5.3	Data Read .....	72
6.5.4	Data Write .....	74
6.5.5	Erase & Write Protect Management .....	75
6.5.6	Read CID/CSD Registers.....	76
6.5.7	Reset Sequence .....	76
6.5.8	Clock Control.....	76
6.5.9	Error Conditions .....	77
6.5.9.1	CRC and Illegal Command.....	77
6.5.9.2	Read, Write and Erase Time-out Conditions .....	77
6.5.9.3	Read ahead in Multiple Block read operation .....	78
6.5.10	Memory Array Partitioning.....	78
6.6	SPI Mode Transaction Packets .....	79
6.6.1	Command Tokens.....	79
6.6.1.1	Command Format.....	79
6.6.1.2	Command Classes.....	79
6.6.1.3	Detailed Command Description .....	80
6.6.2	Responses.....	83
6.6.2.1	Format R1 .....	83
6.6.2.2	Format R1b.....	83
6.6.2.3	Busy.....	83
6.6.2.4	Format R2.....	84
6.6.2.5	Format R3.....	85
6.6.2.6	Data Response .....	85
6.6.3	Data Tokens .....	86
6.6.4	Data Error Token.....	86
6.6.5	Clearing Status Bits.....	87
6.7	Card Registers.....	88
6.8	SPI Bus Timing Diagrams .....	88
6.8.1	Command/Response .....	89
6.8.1.1	Host Command to Card Response - Card is Ready .....	89
6.8.1.2	Host Command to Card Response - Card is Busy .....	89
6.8.1.3	Card Response to Host Command .....	90
6.8.2	Data Read.....	90
6.8.2.1	Single Block Read.....	90
6.8.2.2	Multiple Block Read- Stop Transmission is Sent Between Blocks.....	90
6.8.2.3	Multiple Block Read- Stop Transmission is Sent Within a Block .....	91

## Table of Contents (continued)

6.8.2.4	Reading the CSD Register.....	91
6.8.3	Data Write.....	91
6.8.3.1	Single Block Write.....	91
6.8.3.2	Multiple Block Write .....	92
6.8.4	Timing Values .....	92
6.9	SPI Electrical Interface.....	93
6.10	SPI Bus Operating Conditions.....	93
6.11	Bus Timing .....	93
Ordering Information and Technical Support.....		95
Ordering Information.....		97
MultiMediaCard.....		97
Technical Support Services.....		98
Direct SanDisk Technical Support .....		98
SanDisk Worldwide Web Site .....		98
SanDisk Sales Offices.....		99
Limited Warranty.....		103
Appendix 1: MultiMediaCard Connectors .....		105
Appendix 2: SanDisk MultiMediaCard Product Revision Summary .....		107

## 1.0 Introduction to the MultiMediaCard

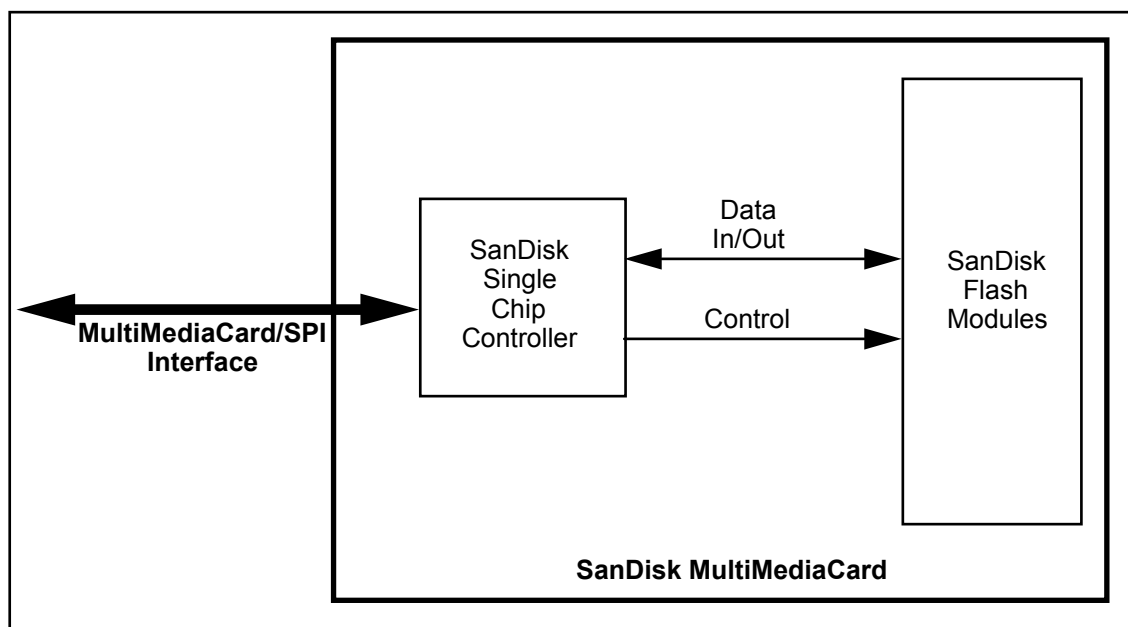
The SanDisk MultiMediaCard is a very small, removable flash storage device, designed specifically for storage applications that put a premium on small form factor, low power and low cost. Flash is the ideal storage medium for portable, battery-powered devices. It features low power consumption and is non-volatile, requiring no power to maintain the stored data. It also has a wide operating range for temperature, shock and vibration.

The MultiMediaCard is well suited to meet the needs of small, low power, electronic devices. With a form factor of 32mm by 24mm and 1.4mm thick, MultiMediaCards are expected to be used in a wide variety of portable devices like mobile phones, pagers and voice recorders. This ultra-small form factor is part of a new, emerging, proposed open standard.

To support this wide range of applications, the MultiMediaCard protocol, a high performance seven pin serial interface, is designed for maximum scalability and configurability. All device and interface configuration data (such as maximum frequency, card identification, etc.) are stored on the card.

The MultiMediaCard interface allows for easy integration into any design, regardless of microprocessor used. For compatibility with existing controllers, the MultiMediaCard offers, in addition to the MultiMediaCard interface, an alternate communication protocol which is based on the Serial Peripheral Interface (SPI) standard.

The MultiMediaCard provides up to 64 million bytes of memory using SanDisk Flash memory chips which were designed by SanDisk especially for use in mass storage applications. In addition to the mass storage specific flash memory chip, the MultiMediaCard includes an on-card intelligent controller which manages interface protocols and data storage and retrieval, as well as Error Correction Code (ECC) algorithms, defect handling and diagnostics, power management and clock control.



**Figure 1-1 MultiMediaCard Block Diagram**

---

## 1.1 Scope

This document describes the key features and specifications of the MultiMediaCard, as well as the information required to interface this product to a host system.

---

## 1.2 Product Models

The MultiMediaCard is available in the capacities shown in the following table:

**Table 1-1 MultiMediaCard Capacities**

Model No.	Capacities
SDMB-8	8 MB
SDMB-16	16 MB
SDMB-32	32 MB
SDMB-48	48 MB
SDMB-64	64 MB

---

## 1.3 System Features

---

### 1.3.1 MultiMediaCard Association Standard Features

- Up to 64 Mbytes of data storage
- MultiMediaCard protocol compatible
- Supports SPI Mode
- Targeted for portable and stationary applications
- Voltage range — Communication: 2.0 - 3.6V, Memory Access: 2.7 - 3.6V
- Maximum data rate with up to 10 cards
- Correction of memory field errors
- Built-in write protection features (permanent and temporary)
- Pre-erase mechanism
- Variable clock rate 0 - 20 Mhz
- Multiple cards stackable on a single physical bus



The performance of the communication channel is described in the table below:

**Table 1-2 MultiMediaCard/SPI Comparison**

MultiMediaCard	SPI
Three-wire serial data bus (Clock, command, data)	Three-wire serial data bus (Clock, dataIn, dataOut) + card specific CS signal.
Up to 64k cards addressable by the bus protocol	Card selection via a hardware CS signal
Easy card identification	Not available
Error-protected data transfer	Optional. A non protected data transfer mode is available.
Sequential and single/multiple block oriented data transfer	*Single/Multiple block read/write

\*Note: Multiple sector Read/Write in SPI mode was approved by the MMCA and is included in MMCA system standard rev 3.1.

---

## 1.4 MultiMediaCard Standard

MultiMediaCards are fully compatible with the MultiMediaCard standard specification listed below:

The MultiMediaCard System Specification  
Version 2.11

This specification may be obtained from:

MultiMediaCard Association  
19672 Stevens Creek Blvd., Suite 404  
Cupertino, CA 95014-2465  
USA  
Phone: 408-253-0441  
Fax: 408-253-8811  
Email: prophet2@mmca.org  
<http://www.mmca.org>

---

## 1.5 Functional Description

SanDisk MultiMediaCards contain a high level, intelligent subsystem as shown in the block diagram, Figure 1-1. This intelligent (microprocessor) subsystem provides many capabilities not found in other types of memory cards. These capabilities include:

1. Host independence from details of erasing and programming flash memory.
2. Sophisticated system for managing defects (analogous to systems found in magnetic disk drives).
3. Sophisticated system for error recovery including a powerful error correction code (ECC).
4. Power management for low power operation.

---

### 1.5.1 Flash Technology Independence

The 512 byte sector size of the MultiMediaCard is the same as that in an IDE magnetic disk drive. To write or read a sector (or multiple sectors), the host computer software simply issues a Read or Write command to the MultiMediaCard. This command contains the address. The host software then waits for the command to complete. The host software does not get involved in the details of how the flash memory is erased, programmed or read. This is extremely important as flash devices are expected to get more and more complex in the future. Because the MultiMediaCard uses an intelligent on-board controller, the host system software will not require changing as new flash memory evolves. In other words, systems that support the MultiMediaCard today will be able to access future SanDisk MultiMediaCards built with new flash technology without having to update or change host software.

---

### 1.5.2 Defect and Error Management

MultiMediaCards contain a sophisticated defect and error management system. This system is analogous to the systems found in magnetic disk drives and in many cases offers enhancements. For instance, disk drives do not typically perform a read after write to confirm the data is written correctly because of the performance penalty that would be incurred. MultiMediaCards do a read after write under margin conditions to verify that the data is written correctly. In the rare case that a bit is found to be defective, MultiMediaCards replace this bad bit with a spare bit within the sector header. If necessary, MultiMediaCards will even replace the entire sector with a spare sector. This is completely transparent to the host and does not consume any user data space.

The MultiMediaCard's soft error rate specification is much better than the magnetic disk drive specification. In the extremely rare case a read error does occur, MultiMediaCards have innovative algorithms to recover the data. This is similar to using retries on a disk drive but is much more sophisticated. The last line of defense is to employ a powerful ECC to correct the data. If ECC is used to recover data, defective bits are replaced with spare bits to ensure they do not cause any future problems.

These defect and error management systems coupled with the solid-state construction give MultiMediaCards unparalleled reliability.

---

### 1.5.3 Endurance

SanDisk MultiMediaCards have an endurance specification for each sector of 100,000 writes (reading a logical sector is unlimited). This is far beyond what is needed in nearly all applications of MultiMediaCards. Even very heavy use of the MultiMediaCard in cellular phones, personal communicators, pagers and voice recorders will use only a fraction of the total endurance over the typical device's five year lifetime. For instance, it would take over 34 years to wear out an area on the MultiMediaCard on which a file of any size (from 512 bytes to capacity) was rewritten 3 times per hour, 8 hours a day, 365 days per year.

With typical applications the endurance limit is not of any practical concern to the vast majority of users.

---

#### 1.5.4 Wear Leveling

SanDisk MultiMediaCards do not require or perform a Wear Level operation.

---

#### 1.5.5 Using the Erase Command

The Erase (sector or group) command provides the capability to substantially increase the write performance of the MultiMediaCard. Once a sector has been erased using the Erase command, a write to that sector will be much faster. This is because a normal write operation includes a separate sector erase prior to write.

---

#### 1.5.6 Automatic Sleep Mode

A unique feature of the SanDisk MultiMediaCard (and other SanDisk products) is automatic entrance and exit from sleep mode. Upon completion of an operation, the MultiMediaCard will enter the sleep mode to conserve power if no further commands are received within 5 msec. The host does not have to take any action for this to occur. In most systems, the MultiMediaCard is in sleep mode except when the host is accessing it, thus conserving power.

When the host is ready to access the MultiMediaCard and it is in sleep mode, any command issued to the MultiMediaCard will cause it to exit sleep and respond.

---

#### 1.5.7 Hot Insertion

Support for hot insertion will be required on the host, but will be supported through the connector. Connector manufacturers will provide connectors that have power pins long enough to be powered before contact is made with the other pins. Please see connector data sheets for more details. This approach is similar to that used in PCMCIA to allow for hot insertion. This applies to both MultiMediaCard and SPI modes.

---

#### 1.5.8 MultiMediaCard Mode

---

##### 1.5.8.1 MultiMediaCard Standard Compliance

The MultiMediaCard is fully compliant with MultiMediaCard Standard Specification, version 2.11. The structure of the Card Specific Data (CSD) register is compliant with CSD structure version 2.11.

---

##### 1.5.8.2 Negotiating Operation Conditions

The MultiMediaCard supports the operation condition verification sequence defined in the MultiMediaCard standard specifications. Should the MultiMediaCard host define an operating voltage range which is not supported by the MultiMediaCard it will put itself in an inactive state and ignore any bus communication. The only way to get the card out of the inactive state is by powering it down and up again.

In addition, the host can explicitly send the card to the inactive state by using the GO\_INACTIVE\_STATE command.

---

#### 1.5.8.3 Card Acquisition and Identification

The MultiMediaCard bus is a single master (MultiMediaCard host) and multi-slaves (cards) bus. The host can query the bus and find out how many cards of which type are currently connected. The MultiMediaCard's CID register is pre-programmed with a unique card identification number which is used during the acquisition and identification procedure.

In addition, the MultiMediaCard host can read the card's CID register using the READ\_CID MultiMediaCard command. The CID register is programmed during the MultiMediaCard testing and formatting procedure, on the manufacturing floor. The MultiMediaCard host can only read this register and not write to it.

---

#### 1.5.8.4 Card Status

MultiMediaCard status is stored in a 32-bit status register which is sent as the data field in the card response to host commands. The Status register provides information about the card's current state and completion codes for the last host command.

The card status can be explicitly read (polled) with the SEND\_STATUS command.

---

#### 1.5.8.5 Memory Array Partitioning

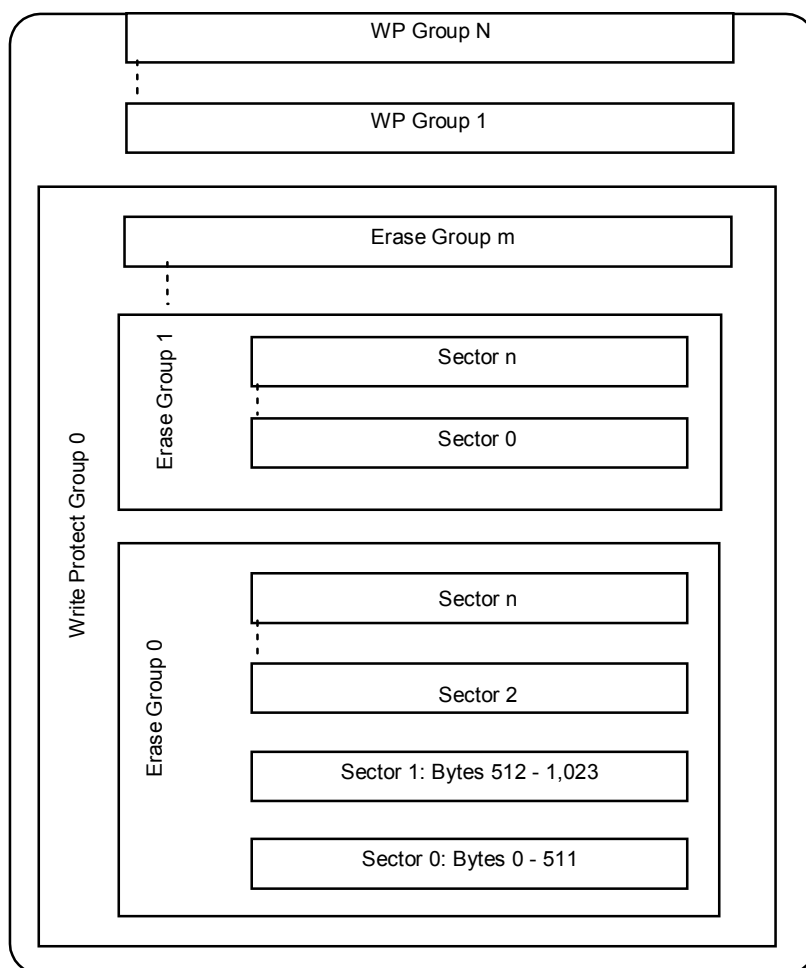
Although the MultiMediaCard memory space is byte addressable with addresses ranging from 0 to the last byte, it is not a simple byte array but divided into several structures.

Memory bytes are grouped into 512 byte blocks called sectors. Every block can be read, written and erased individually.

Sectors are grouped into erase groups of 16 or 32 sectors depending on card size. Any combination of sectors within one group, or any combination of erase groups can be erased in a single erase command. A write command implicitly erases the memory before writing new data into it. An explicit erase command can be used for pre-erasing memory which will speed up the next write operation.

Erase groups are grouped into Write Protect Groups (WPG) of 32 erase groups. The write/erase access to each WPG can be limited individually. A diagram of the memory structure hierarchy is shown in Figure 1-2.

The number of various memory structures, for the different MultiMediaCards are summarized in Table 1-3. The last (highest in address) WPG will be smaller and contain less than 32 erase groups.

**Figure 1-2 Memory Array Partitioning****Table 1-3 Memory Array Structures Summary**

Structure	SDMB-8	SDMB-16	SDMB-32	SDMB-48	SDMB-64
Bytes	8 MB	16 MB	32 MB	48 MB	64 MB
Sector	15,680	31,360	62,720	93,952	125,440
Erase Group Size [sectors]	16	32	32	32	32
# of Erase Groups	980	980	1,960	2,936	3,920
Write Protect Group Size [erase groups]	32	32	32	32	32
# of Write Protect Groups	31	31	62	92	123

Note: All measurements are in units per card.

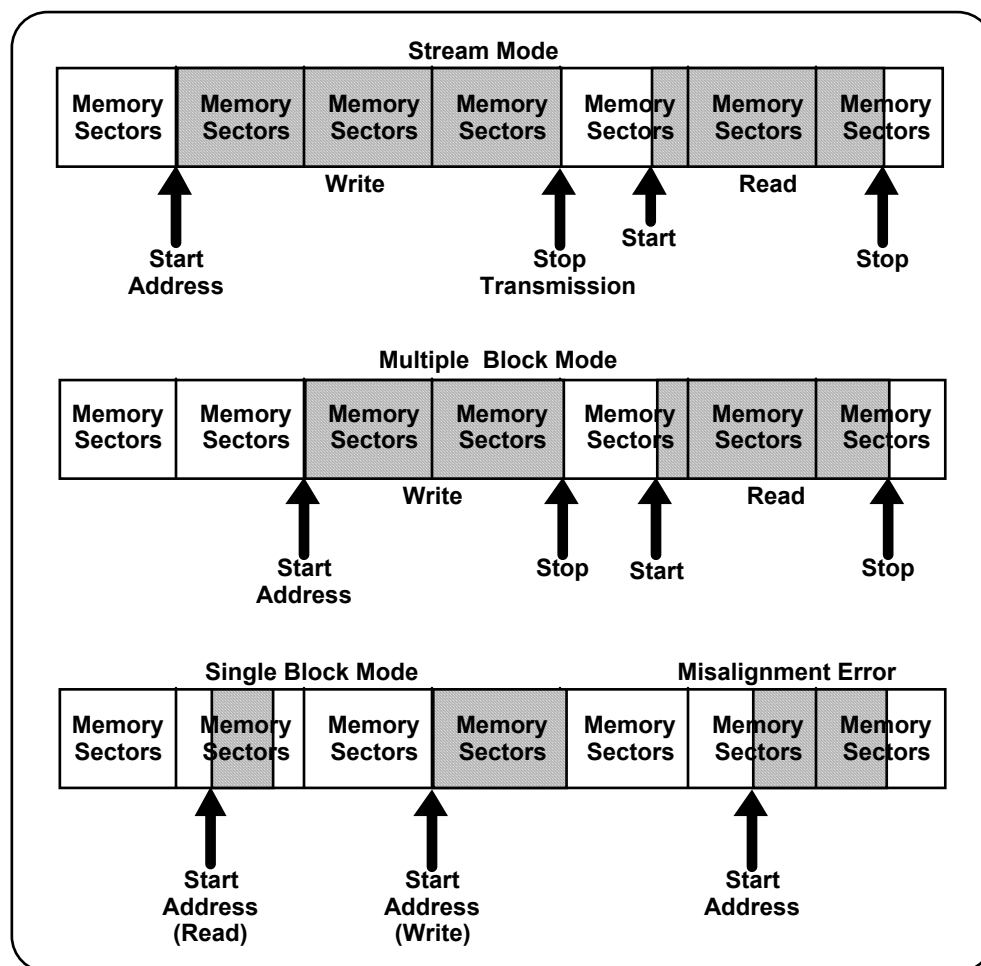


Figure 1-3 Data Transfer Formats

The MultiMediaCard supports three read/write modes as shown in the above figure.

#### Stream Mode

In stream mode the host reads or writes a continuous stream of data. The starting address is specified in the read/write command and the operation ends when the host sends a stop transmission command.

In this mode there is no validity check on the transferred data.

The start address for a read operation can be any random byte address in the valid address space of the memory card. For a write operation, the start address must be sector aligned and the data length must be an integer multiplication of the sector length.

#### Single Block Mode

In this mode, the host reads or writes one data block in a pre-specified length. The data block transmission is protected with 16-bit CRC which is generated by the sending unit and checked by the receiving unit.

The block length for read operations is limited by the device sector size (512 bytes), but can be as small as a single byte. Misalignment is not allowed. Every data block must be contained in a single physical sector.

The block length for write operations must be identical to the sector size and the start address aligned to a sector boundary.

### **Multiple Block Mode**

This mode is similar to the single block mode, but the host can read/write multiple data blocks (all have the same length) which will be stored or retrieved from contiguous memory addresses starting at the address specified in the command.

The operation is terminated with a stop transmission command.

Misalignment and block length restrictions apply to multiple blocks as well, and are identical to the single block read/write operations.

---

#### 1.5.8.7 Data Protection in the Flash Card

Every sector is protected with an Error Correction Code (ECC). The ECC is generated (in the memory card) when the sectors are written and validated when the data is read. If defects are found, the data is corrected prior to transmission to the host.

---

#### 1.5.8.8 Erase

The smallest erasable unit in the MultiMediaCard is a sector. In order to speed up the erase procedure, multiple sectors can be erased at the same time. The erase operation is divided into two stages:

##### **Tagging - Selecting the Sectors for Erasing**

To facilitate selection, a first command with the starting address is followed by a second command with the final address, and all sectors within this range will be selected for erase. After a range is selected, individual sectors (or groups) within that range can be removed using the UNTAG command.

##### **Erasing - Starting the Erase Process**

The sectors are grouped into erase groups of 16 or 32 sectors. Tagging can address sectors or erase groups. Either an arbitrary set of sectors within a single erase group, or an arbitrary selection of erase groups may be erased at one time, but not both together. That is, the unit of measure for determining an erase is either a sector or an erase group. If sectors are tagged, then all selected sectors must lie within the same erase group. Tagging and erasing sectors must follow a strict command sequence.

---

#### 1.5.8.9 Write Protection

Two card level write protection options, permanent and temporary, are available. Both can be set using the PROGRAM\_CSD command. See section 5.3.2, Data Write Format, subsection "CSD Programming". The permanent write protect bit, once set, cannot be cleared. This feature is implemented in the MultiMediaCard controller firmware and not with a physical OTP cell.

---

#### 1.5.8.10 Copy Bit

The content of a MultiMediaCard can be marked as an original or a copy using the copy bit in the CSD register. Once the Copy bit is set (marked as a copy) it cannot be cleared. The Copy bit of the MultiMediaCard is programmed (during test and formatting on the manufacturing floor) as a copy. The MultiMediaCard can be purchased with the copy bit set (copy) or cleared, indicating the card is a master. This feature is implemented in the MultiMediaCard controller firmware and not with a physical OTP cell.

---

#### 1.5.8.11 The CSD Register

All the configuration information of the MultiMediaCard is stored in the CSD register. The MSB bytes of the register contain manufacturer data. The two least significant bytes contain the host controlled data: the card Copy and write protection, the user file format indication, and the user ECC register.

The host can read the CSD register and alter the host-controlled data bytes using the SEND\_CSD and PROGRAM\_CSD commands. See section 5.3.2, Data Write Format, subsection "CSD Programming".

---

### 1.5.9 SPI Mode

The SPI mode is a secondary communication protocol for MultiMediaCards. This mode is a subset of the MultiMediaCard protocol, designed to communicate with an SPI channel, commonly found in Motorola's (and lately a few other vendors') microcontrollers.

---

#### 1.5.9.1 Negotiating Operating Conditions

The operating condition negotiation function of the MultiMediaCard bus is not supported in SPI mode. The host must work within the valid voltage range (2.7 to 3.6) volts of the card.

---

#### 1.5.9.2 Card Acquisition and Identification

The card acquisition and identification function of the MultiMediaCard bus is not supported in SPI mode. The host must know the number of cards currently connected on the bus. Specific card selection is done via the CS signal.

---

#### 1.5.9.3 Card Status

In SPI mode only 16 bits (containing the errors relevant to SPI mode) can be read out of the MultiMediaCard status register.

---

#### 1.5.9.4 Memory Array Partitioning

Memory partitioning in SPI mode is equivalent to MultiMediaCard mode. All read and write commands are byte addressable.



---

#### 1.5.9.5 Read and Write Operations

The SPI mode, as defined in the MMCA Standard, version 2.11, supports only single block read/write. Additionally, the SanDisk MultiMediaCard supports a multiple block read/write which was approved by the MMCA and will be included in a future MultiMediaCard System Specification.

---

#### 1.5.9.6 Data Transfer Rate

Same as for the MultiMediaCard mode when the card is operating in single block read/write mode.

---

#### 1.5.9.7 Data Protection in the MultiMediaCard

Same as for the MultiMediaCard mode.

---

#### 1.5.9.8 Erase

Same as in MultiMediaCard mode.

---

#### 1.5.9.9 Write Protection

Same as in MultiMediaCard mode.

## 2.0 Product Specifications

For all the following specifications, values are defined at ambient temperature and nominal supply voltage unless otherwise stated.

### 2.1 System Environmental Specifications

Temperature	Operating: Non-Operating:	-25° C to 85° C -40° C to 85° C
Humidity	Operating: Non-Operating:	8% to 95%, non-condensing 8% to 95%, non-condensing
ESD Protection	Contact Pads:  Non Contact Pad Area:	± 4kV, Human body model according to ANSI EOS/ESD-S5.1-1998  ± 8kV (coupling plane discharge) ± 15kV (air discharge) Human body model per IEC61000-4-2
Acoustic Noise:		0 dB
Vibration	Operating: Non-Operating:	15 G peak to peak max. 15 G peak to peak max.
Shock	Operating: Non-Operating:	1,000 G max. 1,000 G max.
Altitude (relative to sea level)	Operating: Non-Operating:	80,000 feet max. 80,000 feet max.

### 2.2 Typical System Power Requirements

Operation	@3.3V	@2.7V
Read	<58mA	<58mA
Write	<60mA	<58mA
Sleep	50µA (typical) 150µA (max)	40µA (typical) 120µA (max)

---

## 2.3 System Performance

	Typical	Maximum
Block Read Access Time	1.5 msec	15 msec
CMD1 to Ready (after power up)	50 msec	500 msec
Sleep to Ready	1 msec	2 msec

Notes: All values quoted are under the following conditions:

- a) Voltage range: 2.7 V to 3.6 V.
- b) Temperature range: -25° C to 85° C.
- c) Are independent of the MultiMediaCard clock frequency.

---

## 2.4 System Reliability and Maintenance

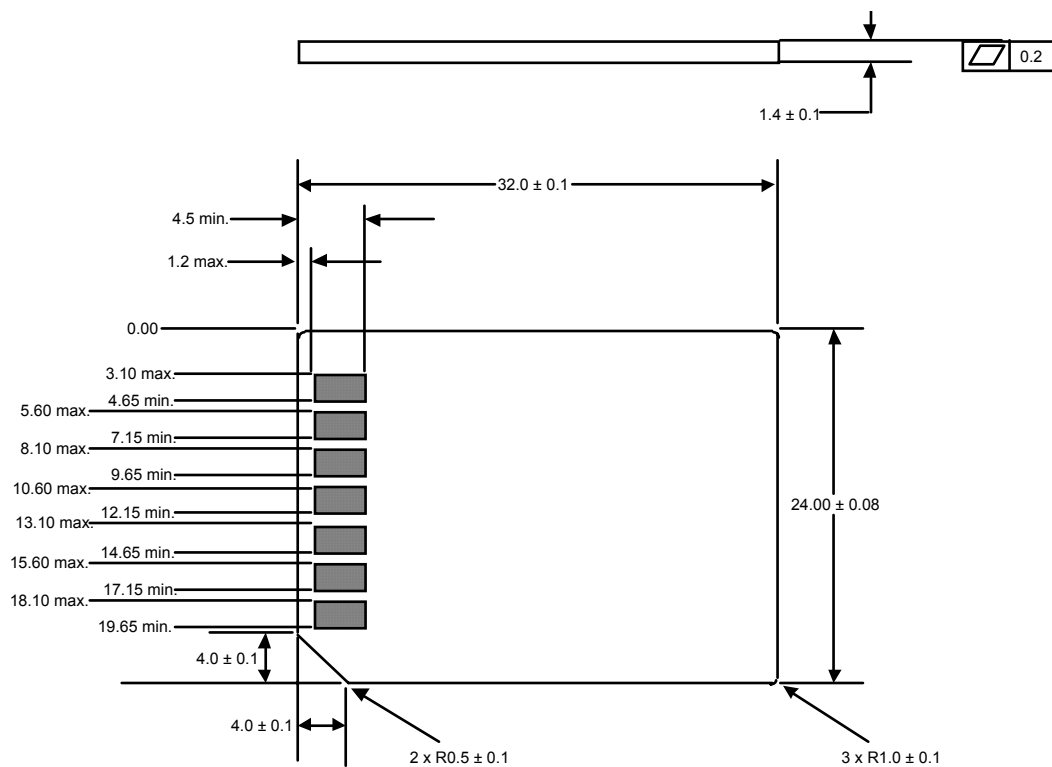
MTBF	> 1,000,000 hours
Preventive Maintenance	None
Data Reliability	< 1 non-recoverable error in $10^{14}$ bits read
Endurance	100,000 write/erase cycles

---

## 2.5 Physical Specifications

Refer to the following table and to Figure 2-1 for MultiMediaCard physical specifications and dimensions.

Weight	1.5 g. maximum
Length:	32mm $\pm$ 0.1mm
Width:	24mm $\pm$ 0.08mm
Thickness:	1.4mm $\pm$ 0.1mm



All dimensions are in millimeters.

**Figure 2-1 MultiMediaCard Dimensions**

## 3.0 Installation

---

### 3.1 Mounting

The MultiMediaCard can be installed in any platform that has a MultiMediaCard slot and complies with the MultiMediaCard Standard.

## 4.0 MultiMediaCard Interface Description

### 4.1 Physical Description

The MultiMediaCard has seven exposed contacts on one side. (See Figure 2-1.) The host is connected to the MultiMediaCard using a seven pin connector as shown in the Appendix 1 at the end of this manual.

#### 4.1.1 Pin Assignments in MultiMediaCard Mode

**Table 4-1 MultiMediaCard Pad Definition**

Pin #	Name	Type*	MultiMediaCard Description
1	RSV	NC	Not Connected or Always '1'
2	CMD	I/O/PP/OD	Command/Response
3	VSS1	S	Supply voltage ground
4	VDD	S	Supply voltage
5	CLK	I	Clock
6	VSS2	S	Supply voltage ground
7	DAT[0]	I/O/PP	Data 0

\*Note: S=power supply; I=input; O=output; PP=push-pull; OD=open-drain; NC=not connected.

#### 4.1.2 Pin Assignments in SPI Mode

**Table 4-2 SPI Pad Definition**

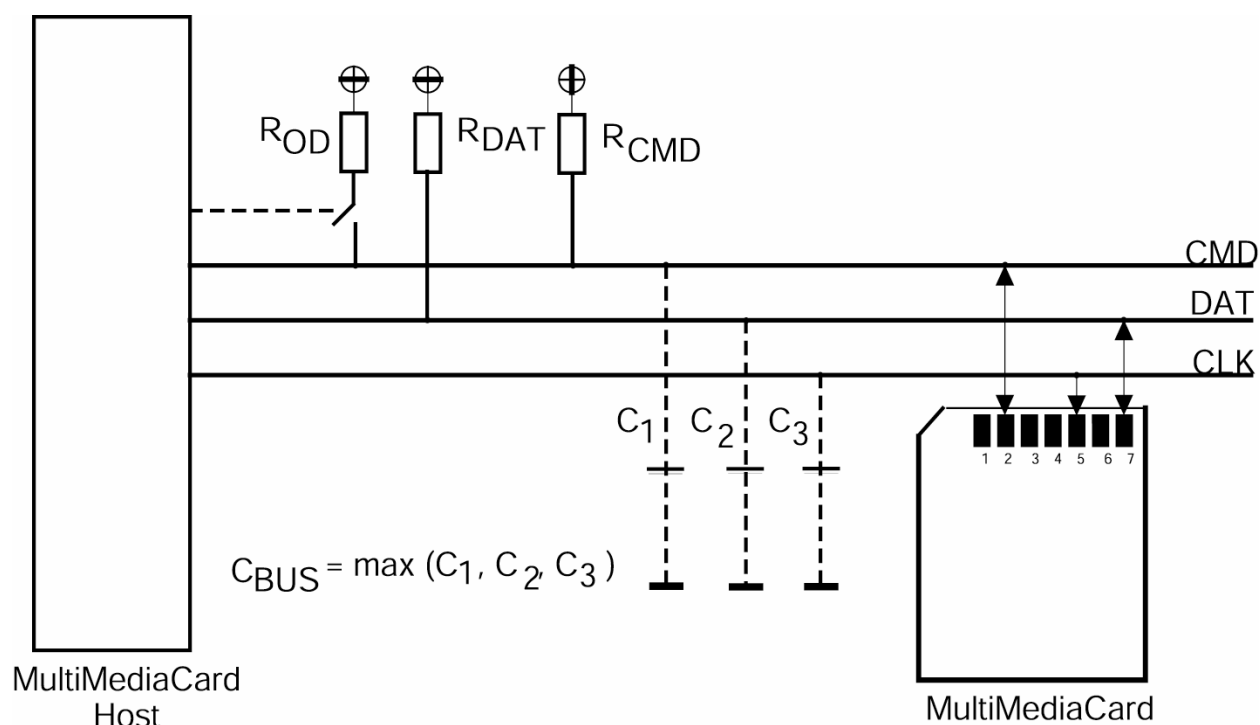
Pin #	Name	Type*	SPI Description
1	CS	I	Chip Select (Active low)
2	DataIn	I	Host to Card Commands and Data
3	VSS1	S	Supply Voltage Ground
4	VDD	S	Supply Voltage
5	CLK	I	Clock
6	VSS2	S	Supply Voltage Ground
7	DataOut	O	Card to Host Data and Status

\*Note: S=power supply; I=input; O=output.

### 4.2 MultiMediaCard Bus Topology

The MultiMediaCard bus has three communication lines and four supply lines:

- CMD: Command is a bi-directional signal. Host and card drivers are operating in two modes, open drain and push pull.
- DAT: Data is a bi-directional signal. Host and card drivers are operating in push pull mode.
- CLK: Clock is a host to card signal. CLK operates in push pull mode.
- VDD: VDD is the power supply line for all cards.
- VSS[1:2]: VSS are two ground lines.



**Figure 4-1 Bus Circuitry Diagram**

The  $R_{OD}$  is switched on and off by the host synchronously to the open-drain and push-pull mode transitions.  $R_{DAT}$  and  $R_{CMD}$  are pull-up resistors protecting the CMD and the DAT line against bus floating when no card is inserted or when all card drivers are in a hi-impedance mode.

A constant current source can replace the  $R_{OD}$  in order to achieve better performance (constant slopes for the signal rising and falling edges). If the host does not allow the switchable  $R_{OD}$  implementation, a fix  $R_{CMD}$  can be used. Consequently the maximum operating frequency in the open drain mode has to be reduced in this case.

#### Hot Insertion/Removal

Hot insertion and removal are allowed. The SanDisk MultiMediaCard will not be damaged by inserting or removing it into the MultiMediaCard bus even when the power is up.

- The inserted card will be properly reset also when CLK carries a clock frequency  $f_{pp}$ .
- Data transfer failures induced by removal/insertion should be detected by the bus master using the CRC codes which suffix every bus transaction.

#### 4.2.1 Power Protection

Cards can be inserted/removed into/from the bus without damage. If one of the supply pins ( $V_{DD}$  or  $V_{SS}$ ) is not connected properly, then the current is drawn through a data line to supply the card.

If the hot insertion feature is implemented in the host, then the host has to withstand a shortcut between  $V_{DD}$  and  $V_{SS}$  without damage.

### 4.3 SPI Bus Topology

The MultiMediaCard SPI interface is compatible with SPI hosts available on the market. As with any other SPI device, the MultiMediaCard SPI channel consists of the following 4 signals:

- CS: Host to card Chip Select signal.
- CLK: Host to card clock signal
- DataIn: Host to card data signal.
- DataOut: Card to host data signal.

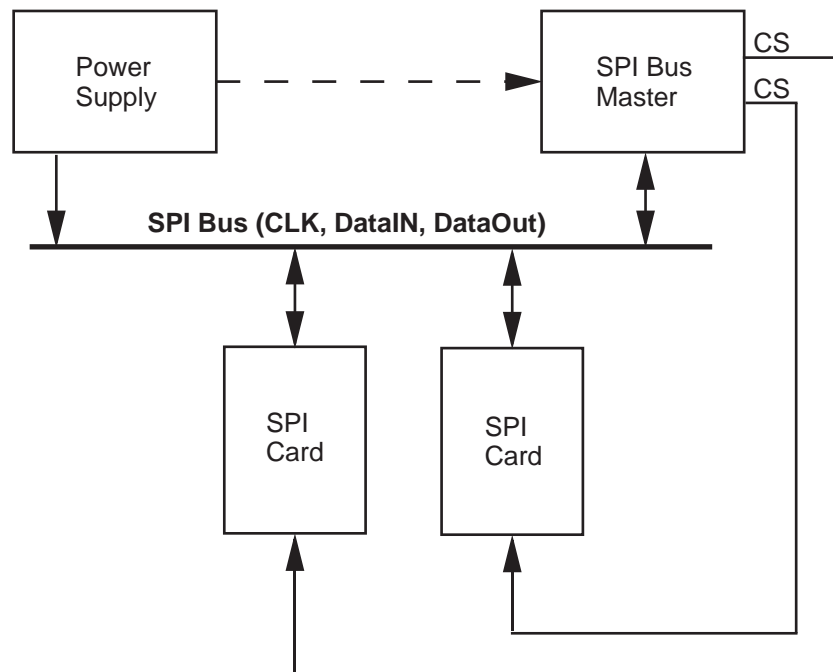
Another SPI common characteristic, which is implemented in the MultiMediaCard as well, is byte transfers. All data tokens are multiples of 8 bit bytes and are always byte-aligned to the CS signal.

The SPI standard defines the physical link only and not the complete data transfer protocol. The MultiMediaCard uses a subset of the MultiMediaCard protocol and command set.

The MultiMediaCard identification and addressing algorithms are replaced by a hardware Chip Select (CS) signal. There are no broadcast commands. A card (slave) is selected for every command, by asserting (active low) the CS signal (see Figure 4-2).

The CS signal must be continuously active for the duration of the SPI transaction (command, response and data). The only exception is card programming time. At this time, the host can de-assert the CS signal without affecting the programming process.

The bi-directional CMD and DAT lines are replaced by uni-directional dataIn and dataOut signals. This prevents command execution while data is being read or written and, therefore, eliminates sequential and multi block read/write operations. Only single block read/write is supported by the SPI channel.



**Figure 4-2 MultiMediaCard Bus System**



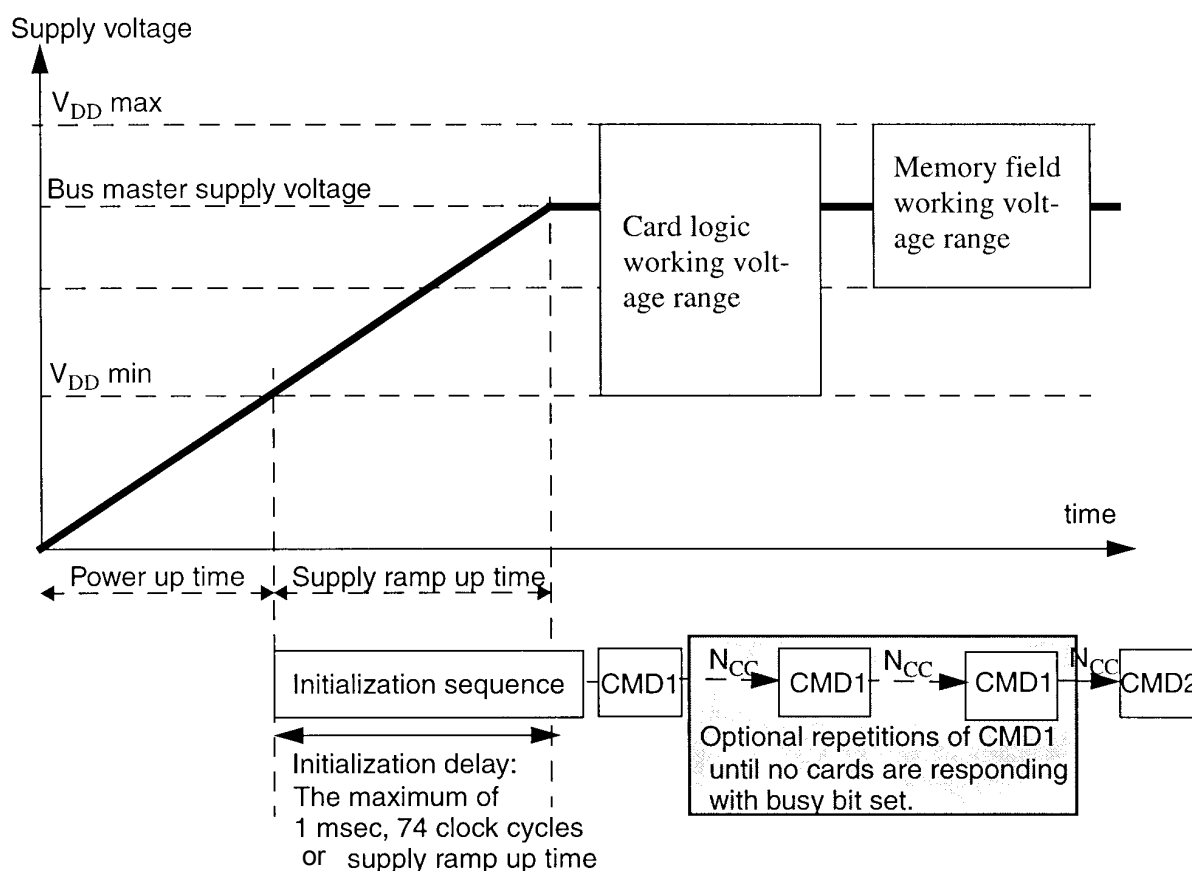
### 4.3.1 Power Protection

Same as for MultiMediaCard mode.

## 4.4 Electrical Interface

### 4.4.1 Power-up

The power up of the MultiMediaCard bus is handled locally in each MultiMediaCard and in the bus master.



**Figure 4-3 Power-up Diagram**

After power-up, including hot insertion (inserting a card when the bus is operating), the MultiMediaCard enters the Idle State. During this state, the MultiMediaCard ignores all bus transactions until CMD1 is received.

CMD1 is a special synchronization command used to negotiate the operation voltage range and to poll the cards until they are out of their power-up sequence. Besides the operation voltage profile of the cards, the response to CMD1 contains a busy flag, indicating that the card is still working on its power-up procedure and is not ready for identification. This bit informs the host that at least one card is not ready. The host has to wait (and continue to poll the cards) until this bit is cleared.

Getting individual cards, as well as the whole MultiMediaCard system, out of Idle State is up to the responsibility of the bus master. Since the power-up time and the supply ramp up time depend on application parameters such as the maximum number of MultiMediaCards, the bus length and the power supply unit, the host must ensure that the power is built up to the operating level (the same level which will be specified in CMD1) before CMD1 is transmitted.

After power-up, the host starts the clock and sends the initializing sequence on the CMD line. This sequence is a contiguous stream of logical ones. The sequence length is the maximum of one msec, 74 clocks or the supply ramp up time. The additional ten clocks (beyond the 64 clocks after which the card should be ready for communication) are provided to eliminate power-up synchronization problems.

#### 4.4.2 Bus Operating Conditions

SPI Mode bus operating conditions are identical to MultiMediaCard Mode bus operating conditions. The CS (chip select) signal timing is identical to the input signal timing. (See Figure 4-5.)

##### General

Parameter	Symbol	Min.	Max.	Unit	Remark
Peak voltage on all lines		-0.5	3.6	V	
<b>All Inputs</b>					
Input Leakage Current		-10	10	μA	
<b>All Outputs</b>					
Output Leakage Current		-10	10	μA	

##### Power supply voltage

Parameter	Symbol	Min.	Max.	Unit	Remark
Supply voltage	V <sub>DD</sub>	2.0	3.6	V	
Supply voltage differentials (V <sub>SS1</sub> , V <sub>SS2</sub> )		-0.5	0.5	V	

The current consumption of any card during the power-up procedure must not exceed 10 mA.

##### Capacitance

Parameter	Symbol	Min.	Max.	Unit	Remark
V <sub>DD</sub> capacitance	C (V <sub>DD</sub> )		3.0	μF	

**Bus Signal Line Load**

The total capacitance  $CL$  of each line of the MultiMediaCard bus is the sum of the bus master capacitance  $CHOST$ , the bus capacitance  $CBUS$  itself, and the capacitance  $CCARD$  of each card connected to this line:

$$CL = CHOST + CBUS + N \cdot CCARD$$

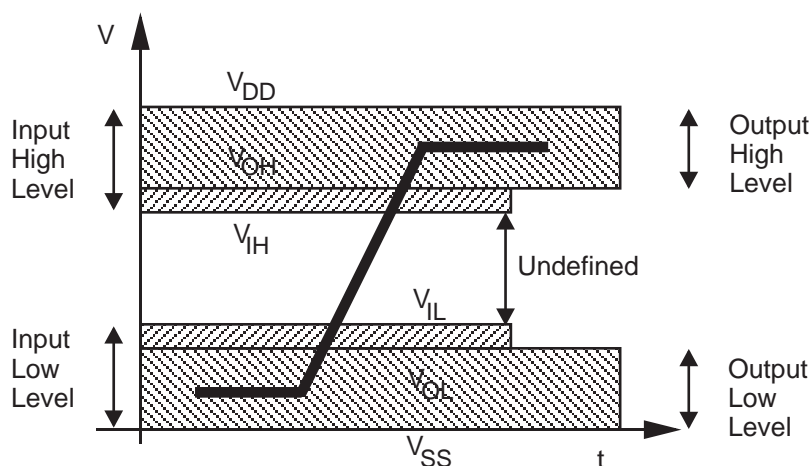
where  $N$  is the number of connected cards. Given the requirement that the sum of the host and bus capacitances not exceed 30 pF for up to 10 cards, and 40 pF for up to 30 cards, the following values must not be exceeded:

Parameter	Symbol	Min.	Max.	Unit	Remark
Pull-up resistance	RCMD RDAT	50	100	k $\Omega$	To prevent bus floating
Bus signal line capacitance	CL		250	pF	fPP # 5 MHz, 30 cards
Bus signal line capacitance	CL		100	pF	fPP # 20 MHz, 10 cards
Single card capacitance	CCARD		7	pF	
Maximum signal line inductance			16	nH	fPP # 20 MHz

---

#### 4.4.3 Bus Signal Levels

Since the bus can be supplied with a variable supply voltage, all signal levels are related to the supply voltage.



**Figure 4-4 Bus Signal Levels**

## 4.4.4 Open-Drain Mode Bus Signal Level

Parameter	Symbol	Min.	Max.	Unit	Conditions
Output HIGH voltage	VOH	$V_{DD}-0.2$		V	$I_{OH} = -100 \mu A$
Output LOW voltage	VOL		0.3	V	$I_{OL} = 2 \text{ mA}$

The input levels are identical with the push-pull mode bus signal levels.

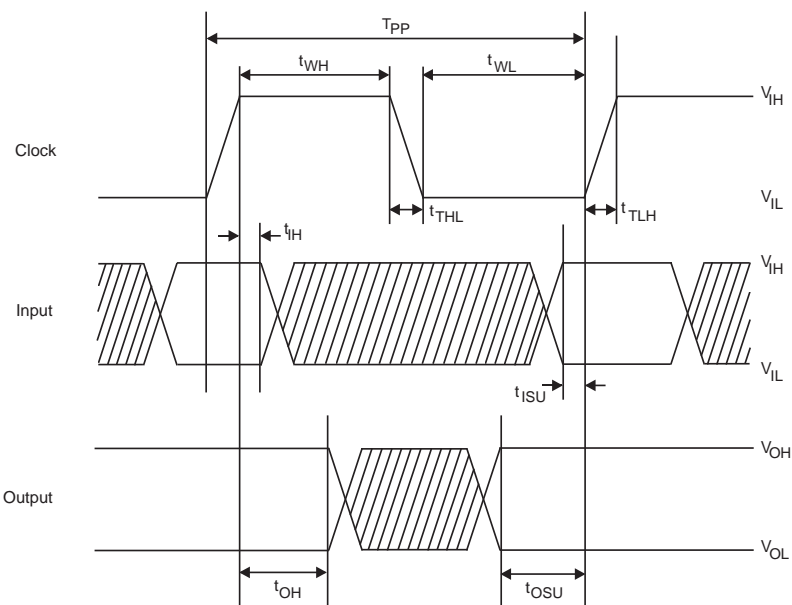
## 4.4.5 Push-pull Mode Bus Signal Level

To meet the requirements of the JEDEC specification JESD8-1A, the card input and output voltages shall be within the following specified ranges for any VDD of the allowed voltage range:

Parameter	Symbol	Min.	Max.	Unit	Conditions
Output HIGH voltage	VOH	$0.75 \cdot V_{DD}$		V	$I_{OH} = -100 \mu A$ @ $V_{DD}(\text{min.})$
Output LOW voltage	VOL		$0.125 \cdot V_{DD}$	V	$I_{OL} = 100 \mu A$ @ $V_{DD}(\text{min.})$
Input HIGH voltage	VIH	$0.625 \cdot V_{DD}$	$V_{DD} + 0.3$	V	
Input LOW voltage	VIL	$V_{SS}-0.3$	$0.25 \cdot V_{DD}$	V	

## 4.4.6 Bus Timing

SanDisk's MultiMediaCards clock data in on the rising edge and out on the falling edge.



Note: Data in the shaded areas is not valid.

**Figure 4-5 Timing Diagram Data Input/Output Referenced to Clock**

**Table 4-3 Bus Timing**

Parameter	Symbol	Min.	Max.	Unit	Remark
<b>Clock CLK (All values are referred to min. (VIH) and max. (VIL))</b>					
<b>Clock Frequency Data Transfer Mode (PP)</b>	$f_{pp}$	0	20	MHz	$C_L \leq 100$ pF (10 cards)
<b>Clock Frequency Identification Mode (OD)</b>	$f_{OD}$	0	400	kHz	$C_L \leq 250$ pF (30 cards)
<b>Clock Low Time</b>	$t_{WL}$	10		ns	$C_L \leq 100$ pF (10 cards)
<b>Clock High Time</b>	$t_{WH}$	10		ns	$C_L \leq 100$ pF (10 cards)
<b>Clock Rise Time</b>	$t_{TLH}$		10	ns	$C_L \leq 100$ pF (10 cards)
<b>Clock Fall Time</b>	$t_{THL}$		10	ns	$C_L \leq 100$ pF (10 cards)
<b>Clock Low Time</b>	$t_{WL}$	50		ns	$C_L \leq 250$ pF (30 cards)
<b>Clock High Time</b>	$t_{WH}$	50		ns	$C_L \leq 250$ pF (30 cards)
<b>Clock Rise Time</b>	$t_{TLH}$		50	ns	$C_L \leq 250$ pF (30 cards)
<b>Clock Fall Time</b>	$t_{THL}$		50	ns	$C_L \leq 250$ pF (30 cards)
<b>Inputs CMD, DAT (referenced to CLK)</b>					
<b>Input set-up time</b>	$t_{ISU}$	3		ns	
<b>Input hold time</b>	$t_{IH}$	3		ns	
<b>Outputs CMD, DAT (referenced to CLK)</b>					
<b>Output set-up time</b>	$t_{OSU}$	5		ns	
<b>Output hold time</b>	$t_{OH}$	5		ns	

## 4.5 MultiMediaCard Registers

There is a set of six registers within the card interface. The OCR, CID and CSD registers carry the card configuration information. The RCA register holds the card-relative communication address for the current session.

### 4.5.1 Operating Conditions Register (OCR)

The 32-bit OCR register stores the  $V_{DD}$  voltage profile of the card. The MultiMediaCard is capable of executing the voltage recognition procedure (CMD1) with any standard MultiMediaCard host using operating voltages from 2 to 3.6 Volts.

Accessing the data in the memory array, however, requires 2.7 to 3.6 Volts. The OCR shows the voltage range in which the card data can be accessed. The structure of the OCR register is described in Table 4-4.

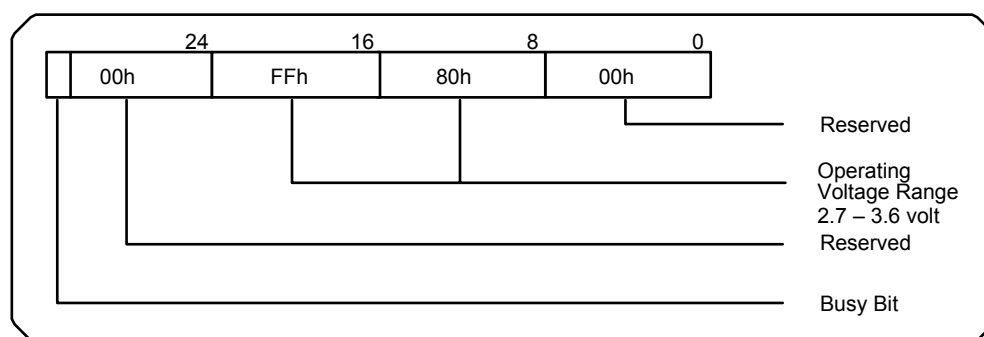
**Table 4-4 OCR Register Definition**

OCR Bit	VDD Voltage Window
0-7	Reserved
8	2.0-2.1
9	2.1-2.2
10	2.2-2.3
11	2.3-2.4
12	2.4-2.5
13	2.5-2.6
14	2.6-2.7
15	2.7-2.8
16	2.8-2.9
17	2.9-3.0
18	3.0-3.1
19	3.1-3.2
20	3.2-3.3
21	3.3-3.4
22	3.4-3.5
23	3.5-3.6
24-30	reserved
31	Card power up status bit (busy)

The level coding of the OCR register is as follows:

- restricted voltage windows=LOW
- card busy=LOW (bit 31)

The least significant 31 bits are constant and will be set as described in Figure 4-6. If set, bit 32, the busy bit, informs the host that the card power up procedure is finished.

**Figure 4-6 OCR Structure**

#### 4.5.2 Card Identification (CID) Register

The CID register is 16 bytes long and contains a unique card identification number as shown in the table below. It is programmed during card manufacturing and can not be changed by MultiMediaCard hosts.

**Table 4-5 CID Fields**

Name	Type	Width	CID - Slice	CID - Value	Comments
Manufacturer ID (MID)	Binary	8	[127:120]	0x02	The manufacturer IDs are controlled and assigned by the MultiMediaCard Association.
OEM/Application ID (OID)	Binary	16	[119:104]	0x0000	Identifies the card OEM and/or the card contents. The OID is assigned by the MMCA. This field may be specifically configured for OEM customers.
Product Name (PNM)	String	48	[103:56]	See Note 1	6 ASCII characters long
Product Revision (PRV)	BCD	8	[55:48]	See Note 2	Two binary coded decimal digits
Serial Number (PSN)	Binary	32	[47:16]		32 Bits unsigned integer
Manufacturing Date Code (MDT)	BCD	8	[15:8]	See Note 3	Manufacturing date – mm/yy (offset from 1997)
CRC7 checksum (CRC)	Binary	7	[7:1]	See Note 4	Calculated
Not used, always '1'		1	[0:0]		

Note 1: Model Name      Name in CD Field  
 SDMB - 8            SDM008  
 SDMB - 16          SDM016  
 SDMB - 32          SDM032  
 SDMB - 48          SDM048  
 SDMB - 64          SDM064

Note 2: The product revision is composed of two Binary Coded Decimal (BCD) digits, four bits each, representing an “n.m” revision number. The “n” is the most significant nibble and the “m” is the least significant nibble. Example: The PRV binary value filed for product revision “6.2” will be: 0110 0010.

Note 3: As an example, this field value for a March 2001 manufacturing date will be: 0011 0100.

Note 4: The CRC Checksum is computed by the following formula:  
 CRC Calculation:  $G(x)=x^7+3+1$   
 $M(x)=(MID-MSB)*x^{119}+...+(CIN-LSB)*x^0$   
 $CRC[6...0]=\text{Remainder}[(M(x)*x^7)/G(x)]$

#### 4.5.3 Card Specific Data (CSD) Register

The CSD register contains all the configuration information required in order to access the card data.

In the table below, the Cell Type column defines the CSD field as Read only (R), One Time Programmable (R/W) or erasable (R/W/E). This table shows, for each field, the value in “real world” units and coded according to the CSD structure. The Model Dependent column marks (with a check mark – √) the CSD fields which are model dependent.

**Table 4-6 CSD Register**

Field	Width [bits]	Cell Type	CSD-slice	CSD Value	CSD Code	Model Dep.	Description
CSD_STRUCTURE	2	R	[127:126]	V1.2	1		<b>CSD Structure</b>
SPEC_VERS	4	R	[125:122]	V2.11	2		<b>MultiMediaCard Specification Version</b>
-	2	R	[121:120]	0	0		<b>Reserved</b>
TAAC	8	R	[119:112]	1.5ms	0x26		<b>Data Read Access-Time-1</b>
NSAC	8	R	[111:104]	0	0		<b>Data Read Access-Time-2 in CLK Cycles (NSAC*100)</b>
TRAN_SPEED	8	R	[103:96]	20MHZ	0x2A		<b>Max. Data Transfer Rate</b>
CCC	12	R	[95:84]	See note	0x03F		<b>Card Command Classes</b>
READ_BL_LEN	4	R	[83:80]	512	9		<b>Max. Read Data Block Length</b>
READ_BL_PARTIAL	1	R	[79:79]	Yes	1		<b>Partial Blocks for Read Allowed</b>
WRITE_BLK_MISALIGN	1	R	[78:78]	No	0		<b>Write Block Misalignment</b>
READ_BLK_MISALIGN	1	R	[77:77]	No	0		<b>Read Block Misalignment</b>
DSR_IMP	1	R	[76:76]	No	0		<b>DSR Implemented</b>
-	2	R	[75:74]	0	0		<b>Reserved</b>
C_SIZE	12	R	[73:62]			√	<b>Device Size (C_SIZE)</b>
VDD_R_CURR_MIN	3	R	[61:59]	60mA	6		<b>Max. Read Current @V<sub>DD</sub> Min.</b>
VDD_R_CURR_MAX	3	R	[58:56]	80mA	6		<b>Max. Read Current @V<sub>DD</sub> Max.</b>
VDD_W_CURR_MIN	3	R	[55:53]	60mA	6		<b>Max. Write Current @V<sub>DD</sub> Min.</b>
VDD_W_CURR_MAX	3	R	[52:50]	80mA	6		<b>Max. Write Current @V<sub>DD</sub> Max.</b>
C_SIZE_MULT	3	R	[49:47]			√	<b>Device Size Multiplier (C_SIZE_MULT)</b>
SECTOR_SIZE	5	R	[46:42]	1	0		<b>Erase Sector Size</b>
ERASE_GRP_SIZE	5	R	[41:37]			√	<b>Erase Group Size</b>
WP_GRP_SIZE	5	R	[36:32]	32	0x1F		<b>Write Protect Group Size</b>

Note: Group write protect, Application specific, card lock and I/O command classes are not supported.



WP_GRP_ENABLE	1	R	[31:31]	No	0		<b>Write Protect Group Enable</b>
DEFAULT_ECC	2	R	[30:29]	None	0		<b>Manufacturer Default ECC</b>
R2W_FACTOR	3	R	[28:26]	1:16	4		<b>Read to Write Speed Factor</b>
WRITE_BL_LEN	4	R	[25:22]	512	9		<b>Max. Write Data Block Length</b>
WRITE_BL_PARTIAL	1	R	[21:21]	No	0		<b>Partial Blocks for Write Allowed</b>
-	5	R	[20:16]	0	0		<b>Reserved</b>
FILE_FORMAT_GRP	1	R/W	[15:15]	0	0		<b>Indicates File Format of Selected Group</b>
COPY	1	R/W	[14:14]	Copy	1		<b>Copy Flag (OTP)</b>
PERM_WRITE_PROTECT	1	R/W	[13:13]	No	0		<b>Permanent Write Protection</b>
TMP_WRITE_PROTECT	1	R/W/E	[12:12]	No	0		<b>Temporary Write Protection</b>
FILE_FORMAT	2	R/W	[11:10]	0	0		<b>File Format of Card</b>
ECC	2	R/W/E	[9:8]	None	0		<b>ECC Code</b>
CRC	7	R/W/E	[7:1]			√	<b>CRC</b>
-	1	-	[0:0]	1	1		<b>Not Used, Always '1'</b>

The following sections describe the CSD fields and the relevant data types. If not explicitly defined otherwise, all bit strings are interpreted as binary coded numbers starting with the left bit first.

**CSD\_STRUCTURE**—describes the version of the CSD structure.

**Table 4-7 CSD Register Structure**

<b>CSD_STRUCTURE</b>	<b>CSD Structure Version</b>	<b>Valid for MultiMediaCard Protocol Version</b>
0	CSD version No. 1.0	MultiMediaCard protocol version 1.0-1.2
1	CSD version No. 1.1	MultiMediaCard protocol version 1.4 – 2.2
2 - 3	Reserved	Reserved

**MMC\_PROT**—Defines the MultiMediaCard protocol version supported by the card. It includes the definition of the command set and the card responses. The card identification procedure is compatible for all protocol versions.

**Table 4-8 MultiMediaCard Protocol Version**

<b>SPEC_VERS</b>	<b>MultiMediaCard Protocol Version</b>
0	MultiMediaCard protocol version 1.0-1.2
1	MultiMediaCard protocol version 1.4
2	MultiMediaCard protocol version 2.0 – 2.2
3-15	Reserved

**TAAC**—Defines the asynchronous part (relative to the MultiMediaCard clock (CLK)) of the read access time.

**Table 4-9 TAAC Access Time Definition**

<b>TAAC Bit Position</b>	<b>Code</b>
2:0	time exponent 0=1ns, 1=10ns, 2=100ns, 3=1µms, 4=10µms, 5=100µms, 6=1ms, 7=10ms
6:3	time mantissa 0=reserved, 1=1.0, 2=1.2, 3=1.3, 4=1.5, 5=2.0, 6=2.5, 7=3.0, 8=3.5, 9=4.0, A=4.5, B=5.0, C=5.5, D=6.0, E=7.0, F=8.0
7	reserved

**NSAC**—Defines the worst case for the clock dependent factor of the data access time. The unit for NSAC is 100 clock cycles. Therefore, the maximal value for the clock dependent part of the read access time is 25.5k clock cycles.

The total read access time  $N_{AC}$  as expressed in the Table 5-12 is the sum of TAAC and NSAC. It has to be computed by the host for the actual clock rate. The read access time should be interpreted as a typical delay for the first data bit of a data block or stream from the end bit on the read commands.

**TRAN\_SPEED**—The following table defines the maximum data transfer rate TRAN\_SPEED:

**Table 4-10 Maximum Data Transfer Rate Definition**

<b>TRAN_SPEED Bit</b>	<b>Code</b>
2:0	transfer rate exponent 0=100kbit/s, 1=1Mbit/s, 2=10Mbit/s, 3=100Mbit/s, 4... 7=reserved
6:3	time mantissa 0=reserved, 1=1.0, 2=1.2, 3=1.3, 4=1.5, 5=2.0, 6=2.5, 7=3.0, 8=3.5, 9=4.0, A=4.5, B=5.0, C=5.5, D=6.0, E=7.0, F=8.0
7	reserved

**CCC**—The MultiMediaCard command set is divided into subsets (command classes). The card command class register CCC defines which command classes are supported by this card. A value of '1' in a CCC bit

means that the corresponding command class is supported. For command class definition refer to Table 5-2.

**Table 4-11 Supported Card Command Classes**

CCC Bit	Supported Card Command Class
0	class 0
1	class 1
.....	
11	class 11

**READ\_BL\_LEN**— The data block length is computed as  $2^{\text{READ\_BL\_LEN}}$ . The block length might therefore be in the range 1, 2, 4...2048 bytes:

**Table 4-12 Data Block Length**

READ_BL_LEN	Block Length
0	$2^0 = 1$ Byte
1	$2^1 = 2$ Bytes
.....	
11	$2^{11} = 2048$ Bytes
12-15	reserved

**READ\_BL\_PARTIAL**— Defines whether partial block sizes can be used in block read commands.

READ\_BL\_PARTIAL=0 means that only the READ\_BL\_LEN block size can be used for block-oriented data transfers.

READ\_BL\_PARTIAL=1 means that smaller blocks can be used as well. The minimum block size will be equal to minimum addressable unit (one byte)

**WRITE\_BLK\_MISALIGN**— Defines if the data block to be written by one command can be spread over more than one physical block of the memory device. The size of the memory block is defined in WRITE\_BL\_LEN.

WRITE\_BLK\_MISALIGN=0 signals that crossing physical block boundaries is invalid.

WRITE\_BLK\_MISALIGN=1 signals that crossing physical block boundaries is allowed.

**READ\_BLK\_MISALIGN**— Defines if the data block to be read by one command can be spread over more than one physical block of the memory device. The size of the memory block is defined in READ\_BL\_LEN.

READ\_BLK\_MISALIGN=0 signals that crossing physical block boundaries is invalid.

READ\_BLK\_MISALIGN=1 signals that crossing physical block boundaries is allowed.

**DSR\_IMP**— Defines if the configurable driver stage is integrated on the card. If set, a driver stage register (DSR) must be implemented also.

**Table 4-13 DSR Implementation Code Table**

DSR IMP	DSR Type
0	no DSR implemented
1	DSR implemented

**C\_SIZE (Device Size)**— This parameter is used to compute the card capacity. The memory capacity of the card is computed from the entries C\_SIZE, C\_SIZE\_MULT and READ\_BL\_LEN as follows:

$$\text{memory capacity} = \text{BLOCKNR} * \text{BLOCK\_LEN}$$

where

$$\text{BLOCKNR} = (\text{C\_SIZE} + 1) * \text{MULT}$$

$$\text{MULT} = 2^{\text{C\_SIZE\_MULT} + 2}$$

$$\text{BLOCK\_LEN} = 2^{\text{READ\_BL\_LEN}}$$

$$(\text{C\_SIZE\_MULT} < 8)$$

$$(\text{READ\_BL\_LEN} < 12)$$

Therefore, the maximum capacity which can be coded is  $4096 * 512 * 2048 = 4$  GBytes. Example: A four MByte card with BLOCK\_LEN = 512 can be coded with C\_SIZE\_MULT = 0 and C\_SIZE = 2047.

**VDD\_R\_CURR\_MIN, VDD\_W\_CURR\_MIN**—The minimum values for read and write currents on VDD power supply are coded as follows:

**Table 4-14 V<sub>DD</sub> Minimum Current Consumption**

VDD_R_CURR_MIN VDD_W_CURR_MIN	Code For Current Consumption @ V <sub>DD</sub>
2:0	0=0.5mA; 1=1mA; 2=5mA; 3=10mA; 4=25mA; 5=35mA; 6=60mA; 7=100mA

**VDD\_R\_CURR\_MAX, VDD\_W\_CURR\_MAX**—The maximum values for read and write currents on VDD power supply are coded as follows:

**Table 4-15 V<sub>DD</sub> Maximum Current Consumption**

VDD_R_CURR_MAX VDD_W_CURR_MAX	Code For Current Consumption @ V <sub>DD</sub>
2:0	0=1mA; 1=5mA; 2=10mA; 3=25mA; 4=35mA; 5=45mA; 6=80mA; 7=200mA

**C\_SIZE\_MULT (Device Size Multiplier)**—This parameter is used for coding a factor MULT for computing the total device size (see 'C\_SIZE'). The factor MULT is defined as  $2^{\text{C\_SIZE\_MULT} + 2}$ .

**Table 4-16 Multiply Factor For The Device Size**

C_SIZE_MULT	MULT
0	$2^2 = 4$
1	$2^3 = 8$
2	$2^4 = 16$
3	$2^5 = 32$
4	$2^6 = 64$
5	$2^7 = 128$
6	$2^8 = 256$
7	$2^9 = 512$

**SECTOR\_SIZE**—The size of an erasable sector. The contents of this register is a 5-bit binary coded value, defining the number of write blocks (see **WRITE\_BL\_LEN**). The actual size is computed by increasing this number by one. A value of zero means 1 write block, 31 means 32 blocks.

**ERASE\_GRP\_SIZE**—The size of an erasable group. The contents of this register is a 5-bit binary coded value, defining the number of sectors (see **SECTOR\_SIZE**). The actual size is computed by increasing this number by one. A value of zero means 1 sector, 31 means 32 sectors.

**WP\_GRP\_SIZE**—The size of a write protected group. The contents of this register is a 5-bit binary coded value, defining the number of Erase Groups (see **ERASE\_GRP\_SIZE**). The actual size is computed by increasing this number by one. A value of zero means 1 erase group, 31 means 32 erase groups.

**WP\_GRP\_ENABLE**—A value of '0' means no group write protection possible.

**DEFAULT\_ECC**—Set by the card manufacturer. It defines the ECC code which is recommended for use. The field definition is the same as for the ECC field described later.

**R2W\_FACTOR**—Defines the typical block program time as a multiple of the read access time. The following table defines the field format.

**Table 4-17 R2W\_FACTOR**

<b>R2W_FACTOR</b>	<b>Multiples of Read Access Time</b>
0	1
1	2 (write half as fast as read)
2	4
3	8
4	16
5	32
6,7	reserved

**WRITE\_BL\_LEN**—Block length for write operations. See **READ\_BL\_LEN** for field coding.

**WRITE\_BL\_PARTIAL**—Defines whether partial block sizes can be used in block write commands.

**WRITE\_BL\_PARTIAL**=‘0’ means that only the **WRITE\_BL\_LEN** block size can be used for block oriented data write.

**WRITE\_BL\_PARTIAL**=‘1’ means that smaller blocks can be used as well. The minimum block size is one byte.

**FILE\_FORMAT\_GROUP**—Indicates the selected group of file formats. This field is read-only for ROM. The usage of this field is shown in Table 4-18.

**COPY**—This bit marks the card as an original (‘0’) or non-original (‘1’). Once set to non-original, this bit cannot be reset to original. The definition of “original” and “non-original” is application dependent and changes no card characteristics.

**PERM\_WRITE\_PROTECT**—Permanently protects the whole card content against overwriting or erasing (all write and erase commands for this card are permanently disabled). The default value is ‘0’, i.e. not permanently write protected.

**TMP\_WRITE\_PROTECT**—Temporarily protects the whole card content from being overwritten or erased (all write and erase commands for this card are temporarily disabled). This bit can be set and reset. The default value is '0', i.e. not write protected.

**FILE\_FORMAT**—Indicates the file format on the card. This field is read-only for ROM. The following formats are defined.

**Table 4-18 FILE\_FORMAT**

FILE_FORMAT_GRP	FILE_FORMAT	Type
0	0	Hard disk-like file system with partition table
0	1	DOS FAT (floppy-like) with boot sector only (no partition table)
0	2	Universal File Format
0	3	Others/Unknown
1	0, 1, 2, 3	Reserved

**ECC**—Defines the ECC code that was used for storing data on the card. This field is used by the host (or application) to decode the user data. The following table defines the field format:

**Table 4-19 ECC Type**

ECC	ECC Type	Maximum Number Of Correctable Bits Per Block
0	none (default)	none
1	BCH (542,512)	3
2-3	reserved	-

**CRC**—The CRC field carries the check sum for the CSD contents. The checksum has to be recalculated by the host for any CSD modification. The default corresponds to the initial CSD contents.

---

#### 4.5.4 Status Register

The MultiMediaCard Status register structure is defined in the following table. The Type and Clear-Condition fields in the table are coded as follows:

Type:

- E - Error bit.
- S - Status bit.
- R - Detected and set for the actual command response.
- X - Detected and set during command execution. The host must poll the card by sending status command in order to read these bits.

Clear Condition:

- A - According to the card current state.
- B - Always related to the previous command. Reception of a valid command will clear it (with a delay of one command).
- C - Clear by read.

**Table 4-20 Status Register**

Bits	Identifier	Type	Value	Description	Clear Condition
31	OUT_OF_RANGE	E R	'0' = no error '1' = error	The commands argument was out of allowed range for this card.	C
30	ADDRESS_ERROR	E R X	'0' = no error '1' = error	A misaligned address, which did not match the block length was used in the command.	C
29	BLOCK_LEN_ERROR	E R	'0' = no error '1' = error	The transferred block length is not valid.	C
28	ERASE_SEQ_ERROR	E R	'0' = no error '1' = error	An error in the sequence of erase commands occurred.	C
27	ERASE_PARAM	E X	'0' = no error '1' = error	An invalid selection, sectors or groups, for erase.	C
26	WP_VIOLATION	E R X	'0' = not protected '1' = protected	The command tried to write a write protected block.	C
25-24	Not applicable. This bit is always set to '0'.				
23	COM_CRC_ERROR	E R	'0' = no error '1' = error	The CRC check of the previous command failed.	B
22	ILLEGAL_COMMAND	E R	'0' = no error '1' = error	Command not legal for the current state	B
21-20	Not applicable. This bit is always set to '0'.				
19	ERROR	E R X	'0' = no error '1' = error	A general or an unknown error occurred during the operation.	C
18	UNDERRUN	E X	'0' = no error '1' = error	The card could not sustain data transfer in stream read mode	C
17	OVERRUN	E X	'0' = no error '1' = error	The card could not sustain data programming in stream write mode	C
16	CID/CSD_OVERWRITE	E R X	'0' = no error '1' = error	Can be one of the following errors: - The CID register has been already written and can not be overwritten. - The read only section of the CSD does not match the card content. - An attempt to reverse the copy (set as original) or permanent WP (unprotect) bits was made.	C
15	WP_ERASE_SKIP	S X	'0' = not protected '1' = protected	Only partial address space was erased due to existing WP blocks.	C
14	Not applicable. This bit is always set to '0'.				
13	ERASE_RESET	S R	'0' = cleared '1' = set	An erase sequence was cleared before executing because an out of erase sequence command was received	C
12-9	CURRENT_STATE	S X	0 = idle 1 = ready 2 = ident 3 = stby 4 = tran 5 = data 6 = rcv 7 = prg 8 = dis 9-15 = reserved	The state of the card when the command was received. If the command execution causes a state change, it will be visible to the host in the response to the next command. The four bits are interpreted as a binary coded number between 0 and 15.	B
8	READY_FOR_DATA	S X	'0' = not ready '1' = ready	Corresponds to buffer empty signaling on the bus. (RDY/BSY)	A
7-0	Reserved. Always set to '0'.				

#### 4.5.5 Relative Card Address (RCA) Register

The 16-bit RCA register carries the card address assigned by the host during the card identification. This address is used for the addressed host-card communication after the card identification procedure. The default value of the RCA register is 0x0001. The value 0x0000 is reserved to set all cards in Stand-by State with CMD7.

#### 4.5.6 MultiMediaCard Registers in SPI Mode

In SPI mode, only the MultiMediaCard CSD and CID registers are accessible. Their format is identical to the format in the MultiMediaCard mode. However, a few fields are irrelevant in SPI mode.

In SPI mode, the card status register has a different, shorter format as well. Refer to the SPI Protocol section for more details.

**Table 4-21 MultiMediaCard Registers in SPI Mode**

Name	Available in SPI Mode	Width (Bytes)	Description
CID	Yes	16	Card identification data (serial number, manufacturer ID etc.)
RCA	No		
DSR	No		
CSD	Yes	16	Card specific data, information about the card operation conditions.
OCR	No		

## 4.6 File System Format

SanDisk MultiMediaCards are formatted with a “hard disk-like” partitioned DOS FAT file system.

Similar to hard disks in PCs, the first data block of the memory consists of a partition table. Thus, using the same notation as for hard disks, i.e. partitioning the memory field into logical sectors of 512 bytes each, the first sector is reserved for this partition table. The data in this sector is structured as follows:

**Table 4-22 Partition Table for Hard Disk-like File System**

Byte Position	Length (bytes)	Entry Description	Value/Range
0x0	446	Consistency Check Routine	
0x1be	16	Partition Table Entry	(See below.)
0x1ce	16	Partition Table Entry	(See below.)
0x1de	16	Partition Table Entry	(See below.)
0x1ee	16	Partition Table Entry	(See below.)
0x1fe	1	Signature	'0x55'
0x1ff	1	Signature	'0xaa'



Every partition entry consists of the following fields:

**Table 4-23 Partition Entry Description**

Byte Position	Length (Bytes)	Entry Description	Value/Range
0x0	1	Boot Descriptor	0x00 (Non-bootable Device), 0x80 (Bootable Device)
0x1	3	First Partition Sector	Address of First Sector
0x4	1	File System Descriptor	0 = Empty 1 = DOS 12-bit FAT < 16 MB 4 = DOS 16-bit FAT < 32 MB 5 = Extended DOS 6 = DOS 16-bit FAT >= 32 MB 0x10-0xff = Free for other File Systems*
0x5	3	Last Partition Sector	Address of Last Sector
0x8	4	First Sector Position Relative to Beginning of Device	Number of First Sector (Linear Address)
0xc	4	Number of Sectors in Partition	Between one and Maximum Number of Sectors on Device

The descriptors marked by an asterisk are not used in DOS systems. Every DOS partition is based on a 12-bit, 16-bit FAT or VFAT respectively. All sector numbers are stored in Little-Endian format (least significant byte first). The start and end addresses of the partition are given in terms of heads, tracks and sectors, and can therefore be ignored for the MultiMediaCard, since the position of the partition can be determined by the last two entries.

The boot sector is described in the following table:

**Table 4-24 Boot Sector Configuration**

Byte Position	Length (Bytes)	Entry Description	Value/Range
0x0	3	Jump Command	0xeb 0xXX 0x90
0x3	8	OEM Name	XXX
0xb	2	Bytes/Sector	512
0xd	1	Sectors/Cluster	XXX (range: 1 - 64)
0xe	2	Reserved Sectors (Number of reserved sectors at the beginning of the media including the boot sector.)	1
0x10	1	Number of FATs	2
0x11	2	Number of Root Directory Entries	512
0x13	2	Number of Sectors on Media	XXX (Depends on card capacity, if the media has more than 65535 sectors, this field is zero and the 'number of total sectors' is set.)
0x15	1	Media Descriptor	0xf8 (Hard Disk)
0x16	2	Sectors/FAT	XXX
0x18	2	Sectors/Track	32 (No Meaning)
0x1a	2	Number of Heads	2 (No Meaning)
0x1c	4	Number of Hidden Sectors	0
0x20	4	Number of Total Sectors	XXX (Depends on Capacity)
0x24	1	Drive Number	0x80
0x25	1	Reserved	0
0x26	1	Extended Boot Signature	0x29
0x27	4	Volume ID or Serial Number	XXX
0x2b	11	Volume Label	XXX (ASCII characters padded with blanks if less than 11 characters.)
0x36	8	File System Type	XXX (ASCII characters identifying the file system type FAT12 or FAT16.)
0x3e	448	Load Program Code	XXX
0x1fe	1	Signature	0x55
0x1ff	1	Signature	0xaa

All 'X' entries are denoting card dependent or non-fixed values. The number of sectors per track and the number of heads are meaningless for the MultiMediaCard and can be ignored.

## 5.0 MultiMediaCard Protocol Description

### 5.1 General

All communication between the host and MultiMediaCards is controlled by the host (master). The host sends commands of two types: broadcast and addressed (point-to-point) commands.

- Broadcast Commands

Broadcast commands are intended for all MultiMediaCards. Some of these commands require a response.

- Addressed (Point-to-Point) Commands

The addressed commands are sent to the addressed MultiMediaCard and cause a response from this card.

A general overview of the command flow is shown in Figure 5-1 for the Card Identification Mode and in Figure 5-2 for the Data Transfer Mode. The commands are listed in the command tables (Table 5-3 through Table 5-11). The dependencies between the current MultiMediaCard state, received command and following state are listed in Table 5-12. In the following sections, the different card operation modes will be described first. Thereafter, the restrictions for controlling the clock signal are defined. All MultiMediaCard commands together with the corresponding responses, state transitions, error conditions and timings are presented in the following sections.

Three operation modes are defined for MultiMediaCards:

- Card Identification Mode

The host will be in card identification mode after reset and while it is looking for new cards on the bus. MultiMediaCards will be in this mode after reset until the SET\_RCA command (CMD3) is received.

- Interrupt Mode

The Interrupt Mode option defined in the MultiMediaCard Standard is not implemented on the SanDisk MultiMediaCard.

- Data Transfer Mode

MultiMediaCards will enter data transfer mode once an RCA is assigned to them. The host will enter data transfer mode after identifying all the MultiMediaCards on the bus.

The following table shows the dependencies between bus modes, operation modes and card states. Each state in the MultiMediaCard state diagram (Figure 5-1 and Figure 5-2) is associated with one bus mode and one operation mode:

**Table 5-1 Bus Modes Overview**

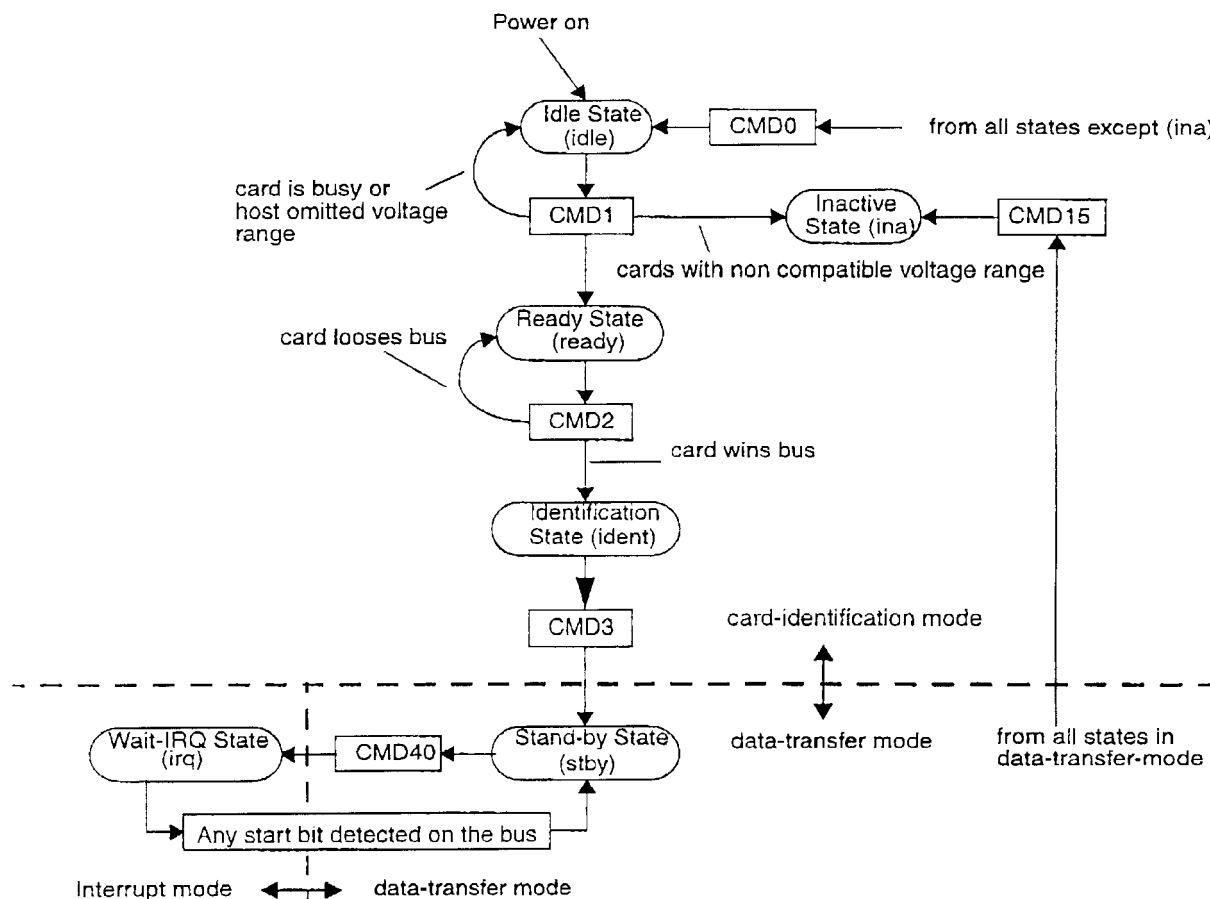
Card State	Operation Mode	Bus Mode
Inactive State	Inactive	Open-Drain
Idle State	Card Identification Mode	
Ready State		
Identification State		
Stand-by State	Data Transfer Mode	Push-Pull
Transfer State		
Sending-data State		
Receive-data State		
Programming State		
Disconnect State		

If a command with improper CRC was received, it is ignored. If there was a command execution (e.g. continuous data read) the card continues in the operation until it gets a correct host command.

---

## 5.2 Card Identification Mode

All the data communication in the Card Identification Mode uses only the command line (CMD).



**Figure 5-1 MultiMediaCard State Diagram (Card Identification Mode)**

### 5.2.1 Reset

GO\_IDLE\_STATE (CMD0) is the software reset command and sets all MultiMediaCards to Idle State regardless of the current card state. MultiMediaCards in Inactive State are not affected by this command.

After power-on by the host, all MultiMediaCards are in Idle State, including the cards that were in Inactive State. Note that at least 74 clock cycles are required prior to starting bus communication.

After power-on or CMD0, all MultiMediaCards' output bus drivers are in a high-impedance state. The host drives the bus at the identification clock rate  $f_{OD}$  (generated by a push-pull driver stage).

### 5.2.2 Operating Voltage Range Validation

The MultiMediaCard standard requires that all MultiMediaCards will be able to establish communication with the host using any operating voltage between  $V_{DD-min}$  and  $V_{DD-max}$ . However, during data transfer minimum and maximum values for  $V_{DD}$  are defined in the CSD and may not cover the whole range. MultiMediaCard hosts are expected to read the card's CSD register and select proper  $V_{DD}$  values or reject the card.

MultiMediaCards that store the CID and CSD data in the payload memory can communicate this information only under data-transfer  $V_{DD}$  conditions. This means if host and card have non compatible  $V_{DD}$  ranges, the card will not be able to complete the identification cycle, nor to send CSD data.

SEND\_OP\_COND (CMD1) is designed to provide MultiMediaCard hosts with a mechanism to identify and reject cards which do not match the host's desired  $V_{DD}$  range. This is accomplished by the host sending the required  $V_{DD}$  voltage window as the operand of this command. MultiMediaCards which cannot perform data transfer in the specified range must discard themselves from further bus operations and go into Inactive State. All other MultiMediaCards will respond concurrently (same method as card identification) sending back their  $V_{DD}$  range. The wired-or result of the response will show all voltage ranges which some of the cards do not support.

By omitting the voltage range in the command, the host can query the MultiMediaCard stack and determine if there are any non-compatibilities before sending out-of-range cards into the Inactive State. Bus query should be used if the host can select a common voltage range or wants to notify the application of non usable cards in the stack.

The busy bit in the CMD1 response can be used by a card to tell the host that it is still working on its power-up/reset procedure (e.g. downloading the register information from memory field) and is not ready yet for communication. In this case the host must repeat CMD1 until the busy bit is cleared.

During the initialization procedure, the host is not allowed to change the OCR values. Changes in the OCR content will be ignored by the MultiMediaCard. If there is a real change in the operating conditions the host must reset the card stack (using CMD0) and begin the initialization procedure once more.

GO\_INACTIVE\_STATE (CMD15) can also be used to send an addressed MultiMediaCard into the Inactive State. This command is used when the host explicitly wants to deactivate a card (e.g. host is changing  $V_{DD}$  into a range which is known to be not supported by this card).

---

### 5.2.3 Card Identification Process

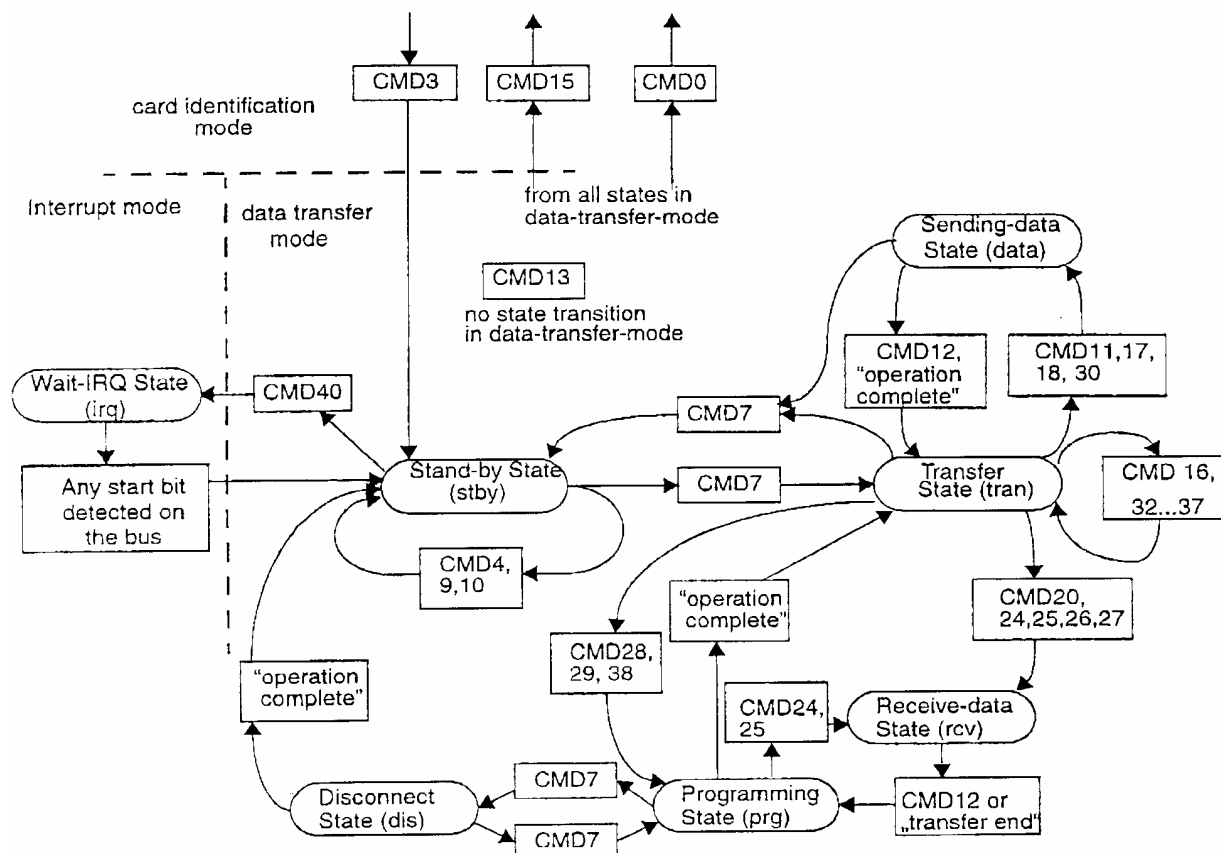
The host starts the card identification process in open-drain mode with the identification clock rate  $f_{OD}$ . The open drain driver stages on the CMD line allow parallel card operation during card identification.

After the bus is activated and a valid operation condition is obtained, the host then asks all cards for their unique card identification (CID) number with the broadcast command ALL\_SEND\_CID (CMD2). All remaining unidentified cards (i.e. those which are in Ready State) simultaneously start sending their CID numbers serially, while bit-wise monitoring their outgoing bit stream. Those cards, whose outgoing CID bits do not match the corresponding bits on the command line in any one of the bit periods, stop sending their CID immediately and must wait for the next identification cycle (cards stay in the Ready State). Since CID numbers are unique for each MultiMediaCard, there should be only one card which successfully sends its full CID-number to the host. This card then goes into Identification State. The host issues CMD3, (SET\_RELATIVE\_ADDR) to assign this card a relative address (RCA), which is shorter than CID and which will be used to address the card in future data transfer mode communication (typically with a higher clock rate than  $f_{OD}$ ). Once the RCA is received the card transfers to the Stand-by State and does not react to further identification cycles. The MultiMediaCard also switches its output drivers from open-drain to push-pull.

The host repeats the identification process as long as it receives a response (CID) to its identification command (CMD2). When no MultiMediaCard responds to this command, all cards have been identified. The time-out condition to recognize completion is the absence of a start bit for more than 5 clock periods after sending CMD2.

### 5.3 Data Transfer Mode

When all cards are in Stand-by State communication over the CMD and DAT lines will be in push-pull mode. Until the content of all CSD registers is known by the host, the  $f_{PP}$  clock rate must remain at  $f_{OD}$  because some cards may have operating frequency restrictions. The host issues SEND\_CSD (CMD9) to obtain the CSD register, e.g. ECC type, block length, card storage capacity, maximum clock rate, etc.



### Figure 5-2 MultiMediaCard State Diagram (Data Transfer Mode)

CMD7 is used to select one MultiMediaCard and place it in the Transfer State. Only one MultiMediaCard can be in the Transfer State at a given time. If a previously selected MultiMediaCard is in the Transfer State, its connection with the host is released and it will move back to the Stand-by State. When CMD7 is issued with the reserved relative card address "0x0000," all cards transfer back to Stand-by State. This command is used to identify new cards without resetting other already acquired cards. MultiMediaCards which already have an RCA do not respond to the identification command flow in this state.

All data communication in the Data Transfer Mode is point-to point between the host and the selected MultiMediaCard (using addressed commands). All addressed commands are acknowledged with a response on the CMD line.

The relationship between the various data transfer modes is summarized in the card state diagram Figure 5-2, and in the following paragraphs:

- All data read commands can be aborted any time by the stop command (CMD12). The data transfer will terminate and the MultiMediaCard will return to the Transfer State. The read commands are: stream read (CMD11), block read (CMD17), multiple block read (CMD18) and send write protect (CMD30).
- All data write commands can be aborted any time by the stop command (CMD12). The write commands must be stopped prior to deselecting the MultiMediaCard by CMD7. The write commands are: stream write (CMD20), block write (CMD24 and CMD25), write CID (CMD26), and write CSD (CMD27).
- If a stream write operation is stopped prior to reaching the block boundary and partial blocks are allowed (as defined in the CSD), the part of the last block will be packed as a partial block and programmed. If partial blocks are not allowed, the data will be discarded.
- As soon as the data transfer is completed, the MultiMediaCard will exit the data write state and move either to the Programming State (transfer is successful) or Transfer State (transfer failed).
- If a block write operation is stopped and the block length and CRC of the last block are valid, the data will be programmed.
- If data transfer in stream write mode is stopped, not byte aligned, the bits of the incomplete byte are ignored and not programmed.
- The MultiMediaCard may provide buffering for stream and block write. This means that the next block can be sent to the card while the previous is being programmed. If all write buffers are full, and as long as the MultiMediaCard is in Programming State (see MultiMediaCard state diagram Figure 5-2), the DAT line will be kept low.
- There is no buffering option for write CSD, write CID, write protection and erase. This means that while the MultiMediaCard is busy servicing any one of these commands, no other data transfer commands will be accepted. DAT line will be kept low as long as the MultiMediaCard is busy and in the Programming State.
- Parameter set commands are *not* allowed while the MultiMediaCard is programming. Parameter set commands are: set block length (CMD16), and erase tagging/untagging (CMD32-37).
- Read commands are *not* allowed while the MultiMediaCard is programming.
- Moving another MultiMediaCard from Stand-by to Transfer State (using CMD7) will not terminate a programming operation. The MultiMediaCard will switch to the Disconnect State and will release the DAT line.
- A MultiMediaCard can be reselected while in the Disconnect State, using CMD7. In this case the MultiMediaCard will move to the Programming State and reactivate the busy indication.
- Resetting a MultiMediaCard (using CMD0 or CMD15) will terminate any pending or active programming operation. This may destroy the data contents on the MultiMediaCard. It is up to the host's responsibility to prevent this.

---

### 5.3.1 Data Read Format

The DAT bus line is high when no data is transmitted. A transmitted data block consists of a start bit (LOW), followed by a continuous data stream. The data stream contains the net payload data (and error correction bits if an off-card ECC is used). The data stream ends with an end bit (HIGH). The data transmission is synchronous to the clock signal.



The payload for block oriented data transfer is preserved by a CRC check sum. The generator polynomial is a standard CCITT polynomial

$$x^{16}+x^{12}+x^5+1.$$

The code is a shortened BCH code with  $d=4$  and is used for payload length of up to 2048 Bytes.

### Stream read

There is a stream-oriented data transfer controlled by READ\_DAT\_UNTIL\_STOP (CMD11). This command instructs the card to send its payload, starting at a specified address, until the host sends a STOP\_TRANSMISSION command (CMD12). Note that the host stop command has an execution delay due to the serial command transmission. The data transfer stops after the end bit of the STOP\_TRANSMISSION command.

If the end of the memory range is reached while sending data and no stop command has yet been sent by the host, the content of the further transferred payload is undefined.

The maximum clock frequency for stream read operation is given by the following formula:

$$\text{max. speed} = \min(\text{TRAN\_SPEED}, (8 * 2^{\text{READ\_BL\_LEN}} - \text{NSAC}) / \text{TAAC})^1$$

If the host attempts to use a higher frequency, the card may not be able to sustain data transfer. Should this happen, the card will set the UNDERRUN error bit in the status register, abort the transmission and wait in the data state for a stop or a new read command.

### Block read

Block read is similar to stream read, except the basic unit of data transfer is a block whose maximum size is defined in the CSD (READ\_BL\_LEN). Smaller blocks whose starting and ending address are wholly contained within one physical block (as defined by READ\_BL\_LEN) may also be transmitted. Unlike stream read, a CRC is appended to the end of each block ensuring data transfer integrity. CMD17 (READ\_SINGLE\_BLOCK) starts a block read and after a complete transfer the card goes back to Transfer State. CMD18 (READ\_MULTIPLE\_BLOCK) starts a transfer of several consecutive blocks. Blocks will be continuously transferred until a stop command is issued.

If the host uses partial blocks whose accumulated length is not block aligned, the card will, at the beginning of the first misaligned block, detect a block misalignment error, set the ADDRESS\_ERROR error bit in the status register, abort transmission and wait (in the *Data State*) for a stop command.

---

## 5.3.2 Data Write Format

The data transfer format is similar to the data read format. For block-oriented write data transfer, the CRC check bits are added to each data block. The card performs a CRC check for each such received data block prior to a write operation. (The polynomial is the same one used for a read operation.) By this mechanism, writing of erroneously transferred data can be prevented.

### Stream write

Stream write (CMD20) means that data is transferred beginning from the starting address until the host issues a stop command. The data stream must start and stop on block boundaries. Since the amount of data to be transferred is not determined in advance, CRC can not be used. If the end of the memory range is reached while sending data and no stop command has been sent by the host, the content of the further transferred payload is discarded.

---

<sup>1)</sup> All upper case names are defined in the CSD.

The maximum clock frequency for stream write operation is given by the following formula:

$$\text{max. speed} = \min(\text{TRAN\_SPEED}, (8 * 2^{\text{WRITE\_BL\_LEN}} - \text{NSAC}) / (\text{TAAC} * \text{R2W\_FACTOR}))$$

If the host attempts to use a higher frequency, the card may not be able to process the data and will stop programming, set the **OVERRUN** error bit in the status register, and while ignoring all further data transfer, wait (in the *Receive-Data-State*) for a stop command. The write operation will also be aborted if the host tries to write over a write-protected area. In this case, however, the card will set the **WP\_VIOLATION** bit.

### Block write

Block write (CMD24 - 27) means that one or more blocks of data are transferred from the host to the card with a CRC appended to the end of each block by the host. If the CRC fails, the card will indicate the failure on the DAT line (see below); the transferred data will be discarded and not written, and all further transmitted blocks (in multiple block write mode) will be ignored.

If the host uses partial blocks whose accumulated length is not block aligned, the card will detect the block misalignment error and abort programming before the beginning of the first misaligned block. The card will set the **ADDRESS\_ERROR** error bit in the status register, and while ignoring all further data transfer, wait (in the *Receive-Data-State*) for a stop command.

The write operation will also be aborted if the host tries to write over a write protected area. In this case, however, the card will set the **WP\_VIOLATION** bit.

After receiving a block of data and completing the CRC check, the card will begin programming and hold the DAT line low if its write buffer is full and unable to accept new data from a new **WRITE\_BLOCK** command. The host may poll the status of the card with a **SEND\_STATUS** command at any time, and the card will respond with its status. The status bit **READY\_FOR\_DATA** indicates whether the MultiMediaCard can accept new data or whether the write process is still in progress. The host may deselect the card by issuing **CMD7** (to select a different card) which will place the card in the Disconnect State and release the DAT line without interrupting the write operation. When reselecting the card, it will reactivate busy indication by pulling DAT to low if programming is still in progress and write buffer is unavailable.

### CSD Programming

Programming of the CSD register does not require a previous block length setting. After sending **CMD27** and getting the **R1** response, send the start bit=0, the modified CSD register = 16Bytes, the **CRC16** = 2Bytes, and the end bit = 1. Only the least significant 16 bits [15:0] of the CSD can be changed by the host. The rest of the CSD register content must match the card CSD. If the card detects a content inconsistency between the old and new CSD register, it will not reprogram the CSD. This is done to ensure validity of the CRC field of the CSD register.

Bits [7:1] are the **CRC7** of bits [127:8] of the CSD register which should be recalculated once the CSD register is changed. After calculating **CRC7**, the **CRC16** should also be calculated for all of the CSD register, [127:0].

### Erase

It is desirable to erase many sectors simultaneously in order to enhance the data throughput. Identification of these sectors is accomplished with the **TAG\_\*** commands. Either an arbitrary set of sectors within a single erase group or an arbitrary selection of erase groups may be erased at one time, but not both together. That is, the unit of measure for determining an erase is either a sector or an erase group, but if a sector, all selected sectors must lie within the same erase group. To facilitate selection, a first command with the starting address is followed by a second command with the final address, and all sectors within this range will be selected for erase. After a range is selected, an individual sector (or group) within that range can be removed using the **UNTAG** command.

The host must adhere to the following command sequence: TAG\_SECTOR\_START, TAG\_SECTOR\_END, UNTAG\_SECTOR (up to 16 untag sector commands can be sent for one erase cycle) and ERASE (or the same sequence for group tagging). The following exception conditions are detected by the MultiMediaCard:

- An erase or tag/untag command is received out of sequence. The card will set the ERASE\_SEQ\_ERROR error bit in the status register and reset the whole sequence.
- An out of sequence command (except SEND\_STATUS) is received. The card will set the ERASE\_RESET status bit in the status register, reset the erase sequence and execute the last command.

If the erase range includes write protected sectors, they will be left intact and only the non protected sectors will be erased. The WP\_ERASE\_SKIP status bit in the status register will be set.

The address field in the tag commands is a sector or a group address in byte units. The card will ignore all LSBs below the group or sector size.

The number of untag commands (CMD34 and CMD37) which are used in a sequence is limited up to 16.

As described above for block write, the MultiMediaCard will indicate that an erase is in progress by holding DAT low.

### Write Protect Management

Card data may be protected against either erase or write by the write protection features. The entire card may be permanently write protected by the manufacturer or content provider by setting the permanent or temporary write protect bits in the CSD. Portions of the data may also be protected (in units of WP\_GRP\_SIZE sectors as specified in the CSD). The SET\_WRITE\_PROT command sets the write protection of the addressed write-protect group, and the CLR\_WRITE\_PROT command clears the write protection of the addressed write-protect group.

The SEND\_WRITE\_PROT command is similar to a single block read command. The card will send a data block containing 32 write protection bits (representing 32 write protect groups starting at the specified address) followed by 16 CRC bits. The address field in the write protect commands is a group address in byte units. The card will ignore all LSBs below the group size.

---

## 5.4 Clock Control

The MultiMediaCard bus clock signal can be used by the MultiMediaCard host to set the cards to energy saving mode or to control the data flow (to avoid under-run or over-run conditions) on the bus. The host is allowed to lower the clock frequency or shut it down.

There are a few restrictions the MultiMediaCard host must follow:

- The bus frequency can be changed at any time (under the restrictions of maximum data transfer frequency, defined by the MultiMediaCard and the identification frequency).
- It is an obvious requirement that the clock must be running for the MultiMediaCard to output data or response tokens. After the last MultiMediaCard bus transaction, the host is required, to provide **eight (8)** clock cycles for the card to complete the operation before shutting down the clock. Following is a list of various MultiMediaCard bus transactions:
  - A command with no response - eight clocks after the host command end bit.
  - A command with response - eight clocks after the card response end bit.
  - A read data transaction - eight clocks after the end bit of the last data block.

- A write data transaction - eight clocks after the CRC status token.
- The host is allowed to shut down the clock of a “busy” card. The MultiMediaCard will complete the programming operation regardless of the host clock. However, the host must provide a clock edge for the card to turn off its busy signal. Without a clock edge the MultiMediaCard (unless previously disconnected by a deselect command -CMD7) will force the DAT line down, permanently.

## 5.5 Cyclic Redundancy Codes (CRC)

The CRC is intended for protecting MultiMediaCard commands, responses and data transfer against transmission errors on the MultiMediaCard bus. One CRC is generated for every command and checked for every response on the CMD line. For data blocks, one CRC per transferred block is generated. The CRC is generated and checked as described in the following.

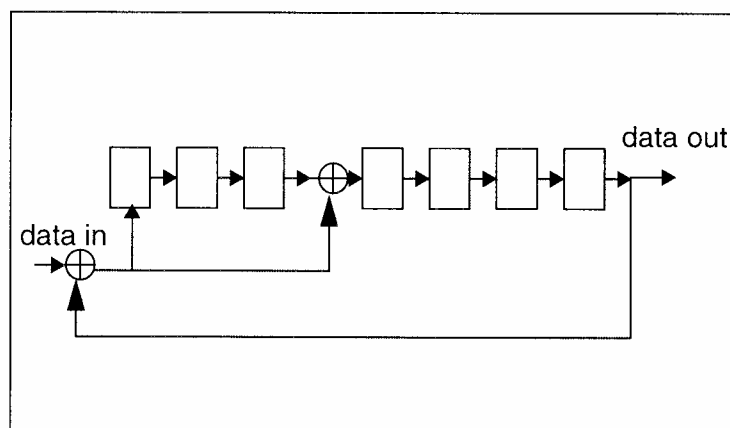
**CRC7**—The CRC7 check is used for all commands, for all responses except type R3, and for the CSD and CID registers. The CRC7 is a 7-bit value and is computed as follows:

generator polynomial:  $G(x) = x^7 + x^3 + 1$ .

$M(x) = (\text{first bit}) * x^n + (\text{second bit}) * x^{n-1} + \dots + (\text{last bit}) * x^0$

$\text{CRC}[6...0] = \text{Remainder} [(M(x) * x^7) / G(x)]$

All CRC registers are initialized to zero. The first bit is the most significant bit of the corresponding bit string (of the command, response, CID or CSD). The degree  $n$  of the polynomial is the number of CRC protected bits decreased by one. The number of bits to be protected is 40 for commands and responses ( $n = 39$ ), and 120 for the CSD and CID ( $n = 119$ ).



**Figure 5-3 CRC7 Generator/Checker**

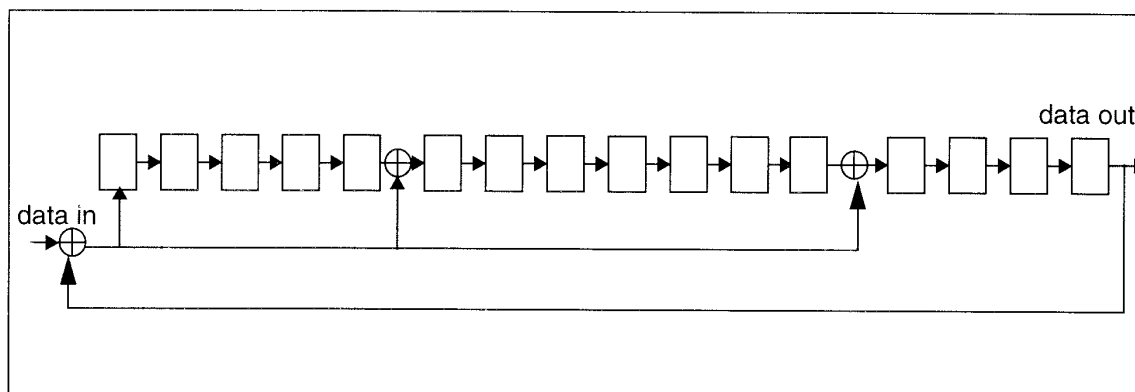
**CRC16**—The CRC16 is used for payload protection in block transfer mode. The CRC check sum is a 16 bit value and is computed as follows:

generator polynomial  $G(x) = x^{16} + x^{12} + x^5 + 1$

$M(x) = (\text{first bit}) * x^n + (\text{second bit}) * x^{n-1} + \dots + (\text{last bit}) * x^0$

$\text{CRC}[15...0] = \text{Remainder} [(M(x) * x^{16}) / G(x)]$

All CRC registers are initialized to zero. The first bit is the first data bit of the corresponding block. The degree  $n$  of the polynomial denotes the number of bits of the data block decreased by one. For example,  $n = 4,095$  for a block length of 512 bytes. The generator polynomial  $G(x)$  is a standard CCITT polynomial. The code has a minimal distance  $d=4$  and is used for a payload length of up to 2,048 bytes ( $n \leq 16,383$ ).



**Figure 5-4 CRC16 Generator/Checker**

## 5.6 Error Conditions

### 5.6.1 CRC and Illegal Command

All commands are protected by CRC (cyclic redundancy check) bits. If the addressed MultiMediaCard's CRC check fails, the card does not respond and the command is not executed. The MultiMediaCard does not change its state, and COM\_CRC\_ERROR bit is set in the status register.

Similarly, if an illegal command has been received, a MultiMediaCard shall not change its state, shall not respond and shall set the ILLEGAL\_COMMAND error bit in the status register. Only the non-erroneous state branches are shown in the state diagrams (Figure 5-1 and Figure 5-2). Table 5-12 contains a complete state transition description.

There are different kinds of illegal commands:

- Commands which belong to classes not supported by the MultiMediaCard (e.g. I/O command CMD39).
- Commands not allowed in the current state (e.g. CMD2 in Transfer State).
- Commands which are not defined (e.g. CMD6).

---

### 5.6.2 Read, Write and Erase Time-out Conditions

The times after which a time-out condition for read/write/erase operations occurs are (card independent) **10 times longer** than the typical access/program times for these operations given below. A card shall complete the command within this time period, or give up and return an error message. If the host does not get a response within the defined time-out it should assume the card is not going to respond any more and try to recover (e.g. reset the card, power cycle, reject, etc.). The typical access and program times are defined as follows:

#### Read

The read access time is defined as the sum of the two times given by the CSD parameters TAAC and NSAC. These card parameters define the typical delay between the end bit of the read command and the start bit of the data block. This number is card dependent and should be used by the host to calculate throughput and the maximal frequency for stream read.

#### Write

The R2W\_FACTOR field in the CSD is used to calculate the typical block program time obtained by multiplying the read access time by this factor. It applies to all write/erase commands (e.g. SET(CLEAR)\_WRITE\_PROTECT, PROGRAM\_CSD(CID) and the block write commands). It should be used by the host to calculate throughput and the maximal frequency for stream write.

#### Erase

The duration of an erase command will be (order of magnitude) the number of sectors to be erased multiplied by the block write delay.

---

## 5.7 Commands

---

### 5.7.1 Command Types

There are four kinds of commands defined on the MultiMediaCard bus:

- Broadcast commands (bc)—sent on CMD, no response
- Broadcast commands with response (bcr)—sent on CMD, response (all cards simultaneously) on CMD
- Addressed (point-to-point) commands (ac)—sent on CMD, response on CMD
- Addressed (point-to-point) data transfer commands (adtc)—sent on CMD, response on CMD, data transfer on DAT

The command transmission always starts with the MSB.

### 5.7.2 Command Format

(Command length 48 bits, 2.4  $\mu$ s @ 20 MHz)

0	1	bit 5...bit 0	bit 31...bit 0	bit 6...bit 0	1
start bit	host	command	argument	CRC7 <sup>1</sup>	end bit

Commands and arguments are listed in Table 5-3 through Table 5-9.

7-bit CRC Calculation:  $G(x) = x^7 + x^3 + 1$

$M(x) = (\text{start bit}) * x^{39} + (\text{host bit}) * x^{38} + \dots + (\text{last bit before CRC}) * x^0$

$\text{CRC}[6...0] = \text{Remainder}[(M(x) * x^7) / G(x)]$

### 5.7.3 Command Classes

The command set of the MultiMediaCard is divided into several classes (See Table 5-2). Each class supports a set of MultiMediaCard functions.

The supported Card Command Classes (CCC) are coded as a parameter in the card specific data (CSD) register of each card, providing the host with information on how to access the card.

**Table 5-2 Card Command Classes (CCCs)**

Card Command Class (CCC)	Class Description	Supported Commands																		
		0	1	2	3	4	7	9	10	11	12	13	15	16	17	18	20			
Class 0	Basic	+	+	+	+	+	+	+	+		+	+	+							
Class 1	Stream Read									+										
Class 2	Block Read													+	+	+				
Class 3	Stream Write																+			
Class 4	Block Write													+						
Class 5	Erase																			
Class 6	Write-Protection <sup>2</sup>																			
Class 7	Lock Card <sup>2</sup>																			
Class 8	Application Specific <sup>2</sup>																			
Class 9	I/O Mode <sup>2</sup>																			
Class 10-11	Reserved																			

<sup>1)</sup> 7-bit Cyclic Redundancy Check.

<sup>2)</sup> Write protection, lock card, application specific, and I/O mode classes are not supported by the SanDisk MultiMediaCard.

Card Command Class (CCC)	Class Description	Supported Commands																							
		24	25	26	27	28	29	30	32	33	34	35	36	37	38	39	40	42	55	56					
Class 0	Basic																								
Class 1	Stream Read																								
Class 2	Block Read																								
Class 3	Stream Write																								
Class 4	Block Write	+	+	+	+																				
Class 5	Erase								+	+	+	+	+	+	+										
Class 6	Write-Protection					+	+	+																	
Class 7	Lock Card																		+						
Class 8	Application Specific																			+	+				
Class 9	I/O Mode																+	+							
Class 10-11	Reserved																								

#### 5.7.4 Detailed Command Description

All future reserved commands have to be 48 bit long, their responses have to be also 48 bits long or they might also have no response.

The following tables define in detail the MultiMediaCard bus commands.

**Table 5-3 Basic Commands and Stream Read Commands (Class 0 And Class 1)**

Cmd Index	Type	Argument	Resp	Abbreviation	Command Description
CMD0	bc	[31:0] don't cares*	-	GO_IDLE_STATE	Resets all cards to Idle State.
CMD1	bcr	[31:0] OCR without busy	R3	SEND_OP_COND	Asks all cards in idle state to send their operation conditions register content in the response on the CMD line.
CMD2	bcr	[31:0] don't cares*	R2	ALL_SEND_CID	Asks all cards to send their CID numbers on the CMD line.
CMD3	ac	[31:16] RCA [15:0] don't cares*	R1	SET_RELATIVE_ADDR	Assigns relative address to the card.
CMD4 <sup>1</sup>	Not Supported				
CMD5	Reserved				
CMD6	Reserved				
CMD7	ac	[31:16] RCA [15:0] don't cares*	R1 (only from the selected card)	SELECT/DESELECT_CARD	Command toggles a card between the Stand-by and Transfer states or between the Programming and Disconnect state.  In both cases the card is selected by its own relative address and deselected by any other address; address 0 deselects all.
CMD8	Reserved				

<sup>1)</sup> The DSR option (as well as the SET\_DSR command) is not supported by the SanDisk MultiMediaCard.



CMD9	ac	[31:16] RCA [15:0] don't cares*	R2	SEND_CSD	Addressed card sends its card-specific data (CSD) on the CMD line.
CMD10	ac	[31:16] RCA [15:0] don't cares*	R2	SEND_CID	Addressed card sends its card identification (CID) on the CMD line.
CMD11	adtc	[31:0] data address <sup>1</sup>	R1	READ_DAT_UNTIL_STOP	Reads data stream from the card, starting at the given address, until a STOP_TRANSMISSION follows.
CMD12	ac	[31:0] don't cares*	R1b <sup>2</sup>	STOP_TRANSMISSION	Terminates a stream or a multiple block read/write operation.
CMD13	ac	[31:16] RCA [15:0] don't cares*	R1	SEND_STATUS	Addressed card sends its status register.
CMD14	Reserved				
CMD15	ac	[31:16] RCA [15:0] don't cares*	-	GO_INACTIVE_STATE	Sets the card to inactive state.

\*Note: The bit places must be filled but the value is irrelevant.

**Table 5-4 Block Read Commands (Class 2)**

Cmd Index	Type	Argument	Resp	Abbreviation	Command Description
CMD16	ac	[31:0] block length	R1	SET_BLOCKLEN	Selects a block length (in bytes) for all following block commands (read and write). <sup>3</sup>
CMD17	adtc	[31:0] data address	R1	READ_SINGLE_BLOCK	Reads a block of the size selected by the SET_BLOCKLEN command. <sup>4</sup>
CMD18	adtc	[31:0] data address	R1	READ_MULTIPLE_BLOCK	Continuously send blocks of data until interrupted by a stop or a new read command.
CMD19	Reserved				

**Table 5-5 Stream Write Commands (Class 3)**

Cmd Index	Type	Argument	Resp	Abbreviation	Command Description
CMD20	adtc	[31:0] data address	R1	WRITE_DAT_UNTIL_STOP	Writes data stream from the host starting at the supplied address, until a STOP_TRANSMISSION follows.
CMD21 .... CMD23	Reserved				

<sup>1)</sup> The addressing capability @ 8 bit address resolution is  $2^{32} = 4$  Gbyte.

<sup>2)</sup> The card may become busy after this command. Refer to Figure 5-18 for more details.

<sup>3)</sup> The default block length is as specified in the CSD (512 bytes). A set block length of less than 512 bytes will cause a write error. The only valid write set block length is 512 bytes. CMD16 is not mandatory if the default is accepted.

<sup>4)</sup> The data transferred must not cross a physical block boundary.

**Table 5-6 Block Write Commands (Class 4)**

Cmd Index	Type	Argument	Resp	Abbreviation	Command Description
CMD24	Adtc	[31:0] data address	R1	WRITE_BLOCK	Writes a block of the size selected by the SET_BLOCKLEN command. <sup>1</sup>
CMD25	Adtc	[31:0] data address	R1	WRITE_MULTIPLE_BLOCK	Continuously writes blocks of data until a STOP_TRANSMISSION follows.
CMD26	Not Applicable				
CMD27	Adtc	[31:0] don't cares*	R1	PROGRAM_CSD	Programming of the programmable bits of the CSD.

\*Note: The bit places must be filled but the value is irrelevant.

**Table 5-7 Write Protection (Class 6)**

Cmd Index	Type	Argument	Resp	Abbreviation	Command Description
CMD28 .... CMD30	MMCA Optional Command, currently not supported.				
CMD31	Reserved				

<sup>1)</sup> All data blocks are responded to with a data response token followed by a busy signal. The data transferred must not cross a physical block boundary.

**Table 5-8 Erase Commands (Class 5)**

Cmd Index	Type	Argument	Resp	Abbreviation	Command Description
CMD32	ac	[31:0] data address	R1	TAG_SECTOR_START	Sets the address of the first sector of the erase group.
CMD33	ac	[31:0] data address	R1	TAG_SECTOR_END	Sets the address of the last sector in a continuous range within the selected erase group, or the address of a single sector to be selected for erase.
CMD34	ac	[31:0] data address	R1	UNTAG_SECTOR	Removes one previously selected sector from the erase selection.
CMD35	ac	[31:0] data address	R1	TAG_ERASE_GROUP_START	Sets the address of the first erase group within a range to be selected for erase.
CMD36	ac	[31:0] data address	R1	TAG_ERASE_GROUP_END	Sets the address of the last erase group within a continuous range to be selected for erase.
CMD37	ac	[31:0] data address	R1	UNTAG_ERASE_GROUP	Removes one previously selected erase group from the erase selection.
CMD38	ac	[31:0] don't cares*	R1b	ERASE	Erases all previously selected sectors or erase groups.

\*Note: The bit places must be filled but the value is irrelevant.

**Table 5-9 I/O Mode Commands (Class 9)**

Cmd Index	Type	Argument	Resp	Abbreviation	Command Description
CMD39 CMD40	MMCA Optional Command, currently not supported.				
CMD41	Reserved				

\*Note: The bit places must be filled but the value is irrelevant.

**Table 5-10 Lock Card Commands (Class 7)**

Cmd Index	Type	Argument	Resp	Abbreviation	Command Description
CMD42 .... CMD54	MMCA Optional Command, currently not supported.				

**Table 5-11 Application Specific Commands (Class 8)**

Cmd Index	Type	Argument	Resp	Abbreviation	Command Description
CMD55 .... CMD56	MMCA Optional Command, currently not supported.				

## 5.8 Card State Transition Table

Table 5-12 defines the MultiMediaCard state transitions in dependency of the received command.

**Table 5-12 Card State Transition Table**

	Current State										
	idle	ready	ident	stby	tran	data	rcv	prg	dis	ina	irq
command	changes to										
class independent											
CRC error	-	-	-	-	-	-	-	-	-	-	stby
command not supported	-	-	-	-	-	-	-	-	-	-	stby
Class 0											
CMD0	idle	idle	idle	idle	idle	idle	idle	idle	idle	-	stby
CMD1, card VDD range compatible	ready	-	-	-	-	-	-	-	-	-	stby
CMD1, card is busy	idle	-	-	-	-	-	-	-	-	-	stby
CMD1, card VDD range not compatible	ina	-	-	-	-	-	-	-	-	-	stby
CMD2, card wins bus	-	ident	-	-	-	-	-	-	-	-	stby
CMD2, card loses bus	-	ready	-	-	-	-	-	-	-	-	stby
CMD3	-	-	stby	-	-	-	-	-	-	-	stby
CMD4	-	-	-	stby	-	-	-	-	-	-	stby
CMD7, card is addressed	-	-	-	tran	-	-	-	-	prg	-	stby
CMD7, card is not addressed	-	-	-	-	stby	stby	-	dis	-	-	stby
CMD9	-	-	-	stby	-	-	-	-	-	-	stby
CMD10	-	-	-	stby	-	-	-	-	-	-	stby
CMD12	-	-	-	-	-	tran	prg	-	-	-	stby
CMD13	-	-	-	stby	tran	data	rcv	prg	dis	-	stby
CMD15	-	-	-	ina	ina	ina	ina	ina	ina	-	stby
Class 1											
CMD11	-	-	-	-	data	-	-	-	-	-	stby
Class 2											
CMD16	-	-	-	-	tran	-	-	-	-	-	stby
CMD17	-	-	-	-	data	-	-	-	-	-	stby
CMD18	-	-	-	-	data	-	-	-	-	-	stby
Class 3											
CMD20	-	-	-	-	rcv	-	-	-	-	-	stby
Class 4											
CMD16	see class 2										
CMD24	-	-	-	-	rcv	-	-	rcv	-	-	stby

CMD25	-	-	-	-	rcv	-	-	rcv	-	-	stby
CMD26	-	-	-	-	rcv	-	-	-	-	-	stby
CMD27	-	-	-	-	rcv	-	-	-	-	-	stby
Class 6											
CMD28...CMD30	MMCA Optional Command, currently not supported										
Class 5											
CMD32	-	-	-	-	tran	-	-	-	-	-	stby
CMD33	-	-	-	-	tran	-	-	-	-	-	stby
CMD34	-	-	-	-	tran	-	-	-	-	-	stby
CMD35	-	-	-	-	tran	-	-	-	-	-	stby
CMD36	-	-	-	-	tran	-	-	-	-	-	stby
CMD37	-	-	-	-	tran	-	-	-	-	-	stby
CMD38	-	-	-	-	prg	-	-	-	-	-	stby
Class 7											
CMD42	MMCA Optional Command, currently not supported.										
Class 8											
CMD-55	MMCA Optional Command, currently not supported.										
CMC-56; RD/WR = 0	MMCA Optional Command, currently not supported.										
CMD-56; RD/WR = 1	MMCA Optional Command, currently not supported.										
Class 9											
CMD39, CMD40	MMCA Optional Command, currently not supported.										
Class 10 - 11											
CMD41...CMD59	Reserved										
CMD60...CMD63	Reserved for manufacturer										

## 5.9 Responses

All responses are sent via the CMD line. The response transmission always starts with the MSB. The response length depends on the response type.

A response always starts with a start bit (always '0'), followed by the bit indicating the direction of transmission (card = '0'). A value denoted by 'x' in the tables below indicates a variable entry. All responses except for the type R3 (see below) are protected by a CRC. Every response is terminated by the end bit (always '1').

There are five types of responses. Their formats are defined as follows:

**R1** (standard response): response length 48 bit.

Bits 45:40 indicate the index of the command which is responded to. The status of the card is coded in 32 bits.

<b>Bit Position</b>	47	46	[45:40]	[39:8]	[7:1]	0
<b>Width (bits)</b>	1	1	6	32	7	1
<b>Value</b>	'0'	'0'	x	x	x	'1'
<b>Description</b>	start bit	transmission bit	command index	card status	CRC7	end bit

**R1b** is identical to R1 with the additional busy signaling via the data.

**R2** (CID, CSD register): response length 136 bits.

The content of the CID register is sent as a response to CMD2 and CMD10. The content of the CSD register is sent as a response to CMD9. Only bits [127...1] of the CID and CSD are transferred, bit [0] of these registers is replaced by the end bit of the response.

<b>Bit Position</b>	135	134	[133:128]	[127:1]	0
<b>Width (bits)</b>	1	1	6	127	1
<b>Value</b>	'0'	'0'	'111111'	x	'1'
<b>Description</b>	start bit	transmission bit	reserved	CID or CSD register incl. internal CRC7	end bit

**R3** (OCR register): response length 48 bits.

The contents of the OCR register is sent as a response to CMD1.

<b>Bit Position</b>	47	46	[45:40]	[39:8]	[7:1]	0
<b>Width (bits)</b>	1	1	6	32	7	1
<b>Value</b>	'0'	'0'	'111111'	x	'1111111'	'1'
<b>Description</b>	start bit	transmission bit	reserved	OCR register	reserved	end bit

**Responses R4 and R5 are not supported.**

### 5.9.1 Data Response

When a data block or the CSD register is written to the card, it will be acknowledged by a CRC Status response. CRC Status response is 5 bits long and has the following format:

[Start bit = 0 | CRC Status 3 bits | End bit = 1]

The CRC Status bits may be:

'010' or '00101 including the Start and End bits' - Data accepted.

'101' or '01011 including the Start and End bits' - Data rejected due to a CRC error.

### 5.10 Timings

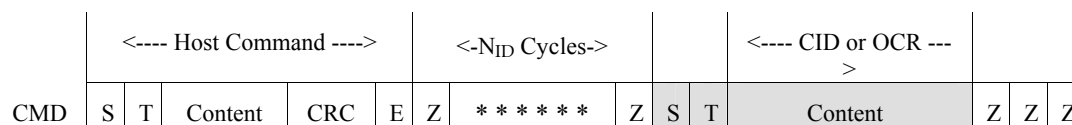
All timing diagrams use the following schematics and abbreviations:

**Table 5-13 Timing Diagram Symbols**

S	Start Bit (= 0)
T	Transmitter Bit (Host = 1, Card = 0)
P	One-cycle Pull-up (= 1)
E	End Bit (=1)
Z	High Impedance State (-> = 1)
D	Data Bits
*	Repeater
CRC	Cyclic Redundancy Check Bits (7 Bits)
	Card Active
	Host Active

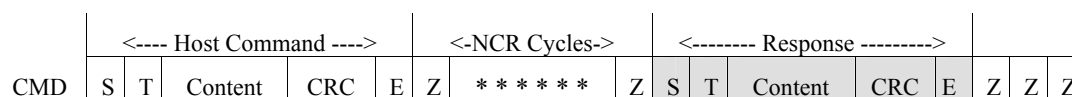
#### 5.10.1 Command and Response

**Card Identification and Card Operation Conditions Timing**—The card identification (CMD2) and card operation conditions (CMD1) timing are processed in the open-drain mode. The card response to the host command starts after exactly  $N_{ID}$  clock cycles.



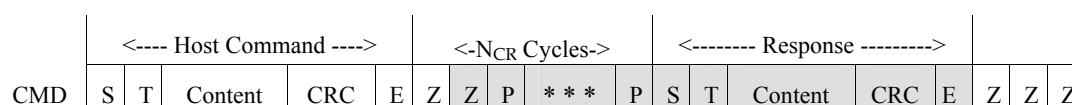
**Figure 5-5 Identification Timing (Card Identification Mode)**

The minimum delay between the host command and card response is  $N_{CR}$  clock cycles. This timing diagram is relevant for host command CMD3.



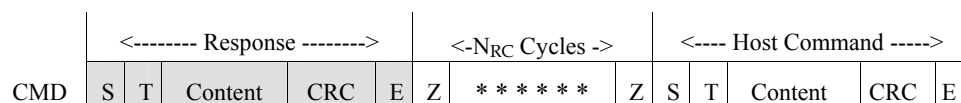
**Figure 5-6 Command Response Timing (Identification Mode)**

There is just one Z bit period followed by P bits pushed up by the responding card. This timing diagram is relevant for all responded host commands except CMD1,2,3.



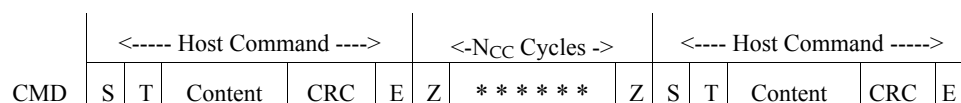
**Figure 5-7 Command Response Timing (Data Transfer Mode)**

**Last Card Response - Next Host Command Timing**—After receiving the last card response, the host can start the next command transmission after at least  $N_{RC}$  clock cycles. This timing is relevant for any host command.



**Figure 5-8 Timing Response End to Next CMD Start (Data Transfer Mode)**

**Last Host Command - Next Host Command Timing Diagram**—After the last command has been sent, the host can continue sending the next command after at least  $N_{CC}$  clock periods. This timing is relevant for any host command that does not have a response.



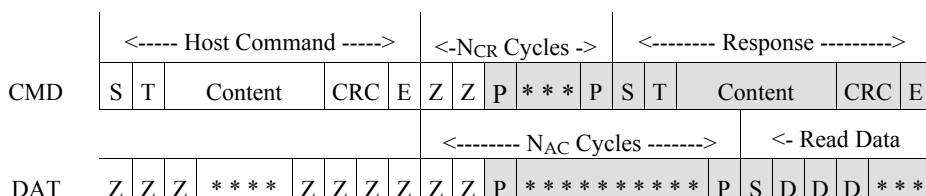
**Figure 5-9 Timing CMD<sub>n</sub> End to CMD<sub>n+1</sub> Start (All Modes)**

In the case where the **CMD<sub>n</sub>** command was a last acquisition command with no further response by any card, then the next **CMD<sub>n+1</sub>** command is allowed to follow after at least  $N_{CC} + 136$  (the length of the R2 response) clock periods.



### 5.10.2 Data Read

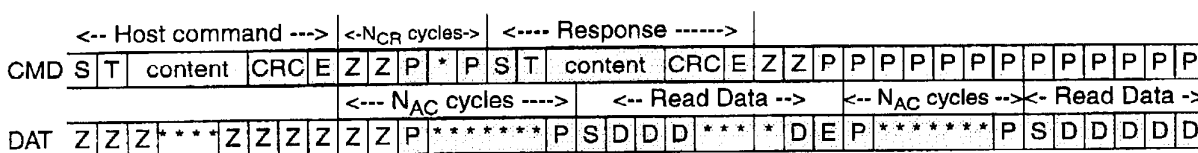
**Single Block Read**—The host selects one card for data read operation by CMD7, and sets the valid block length for block oriented data transfer by CMD16. The basic bus timing for a read operation is given in Figure 5-10. The sequence starts with a single block read command (CMD17) which specifies the start address in the argument field. The response is sent on the CMD line as usual.



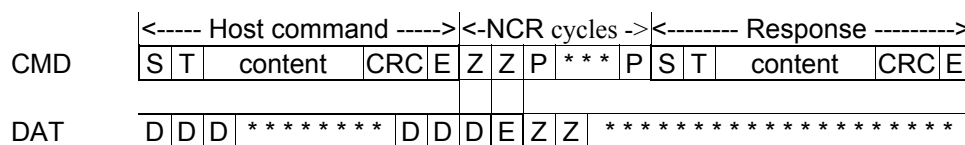
### Figure 5-10 Transfer of Single Block Read

Data transmission from the card starts after the access time delay NAC beginning from the end bit of the read command. After the last data bit, the CRC check bits are suffixed to allow the host to check for transmission errors.

**Multiple Block Read**—In multiple block read mode, the card sends a continuous flow of data blocks following the initial host read command. The data flow is terminated by a stop transmission command (CMD12). Figure 5-11 describes the timing of the data blocks and Figure 5-12 the response to a stop command. The data transmission stops two clock cycles after the end bit of the stop command.



### Figure 5-11 Timing of Multiple Block Read Command



**Figure 5-12 Timing of Stop Command (CMD12, Data Transfer Mode)**

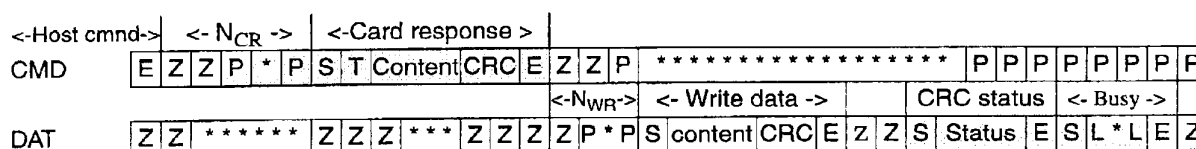
**Stream Read**—The data transfer starts  $N_{AC}$  clock cycles after the end bit of the host command. The bus transaction is identical to that of a read block command (see Figure 5-10). Since the data transfer is not block oriented, the data stream does not include the CRC checksum. Consequently, the host cannot check for data validity. The data stream is terminated by a stop command. The corresponding bus transaction is identical to the stop command for the multiple read block (see Figure 5-12).

**Single Block Write**—The host selects one card for a data write operation by CMD7.

The host sets the valid block length for block-oriented data transfer (a stream write mode is also available) by CMD16.

The basic bus timing for a write operation is given in Figure 5-13. The sequence starts with a single block write command (CMD24) which determines (in the argument field) the start address. It is responded to by the card on the CMD line as usual. The data transfer from the host starts  $N_{WR}$  clock cycles after the card response was received.

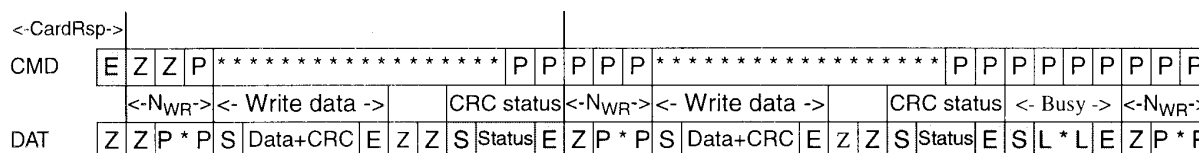
The data is suffixed with CRC check bits to allow the card to check it for transmission errors. The card sends back the CRC check result as a CRC status token on the data line. In the case of transmission error, the card sends a negative CRC status ('101'). In the case of non erroneous transmission, the card sends a positive CRC status ('010') and starts the data programming procedure.



**Figure 5-13 Timing Of The Block Write Command**

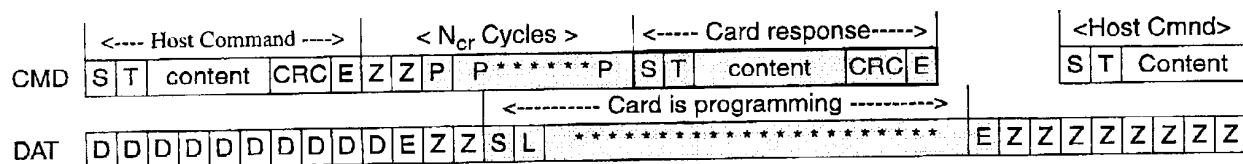
If the MultiMediaCard does not have a free data receive buffer, the card indicates this condition by pulling down the data line to LOW. The card stops pulling down the data line as soon as at least one receive buffer for the defined data transfer block length becomes free. This signaling does not give any information about the data write status which must be polled by the host.

**Multiple Block Write**—In multiple block write mode, the card expects continuous flow of data blocks following the initial host write command. The data flow is terminated by a stop transmission command (CMD12). Figure 5-14 describes the timing of the data blocks with and without card busy signal.



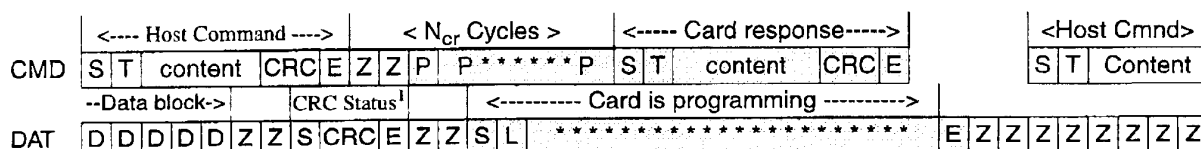
**Figure 5-14 Timing of Multiple Block Write Command**

In write mode, the stop transmission command works similarly to the stop transmission command in the read mode. Figures 5-15 to 5-18 describe the timing of the stop command in different card states.



### Figure 5-15 Stop Transmission During Data Transfer from the Host

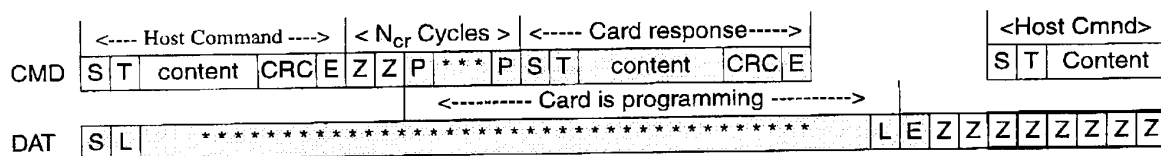
The card will treat a data block as successfully received and ready for programming only if the CRC data of the block was validated and the CRC status token sent back to the host. Figure 5-16 is an example of an interrupted (by a host stop command) attempt to transmit the CRC status block. The sequence is identical to all other stop transmission examples. The end bit of the host command is followed, on the data line, with one more data bit, end bit and two Z clock for switching the bus direction. The received data block, in this case is considered incomplete and will not be programmed.



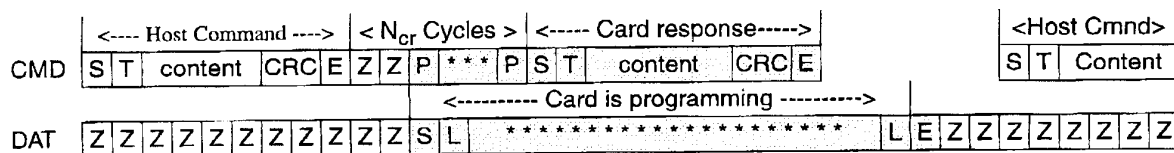
- 1) The card CRC status response was interrupted by the host.

**Figure 5-16 Stop Transmission During CRC Status Transfer from the Card**

All previous examples dealt with the scenario of the host stopping the data transmission during an active data transfer. The following two diagrams describe a scenario of receiving the stop transmission between data blocks. In the first example the card is busy programming the last block while in the second the card is idle. However, there are still unprogrammed data blocks in the input buffers. These blocks are being programmed as soon as the stop transmission command is received and the card activates the busy signal.



**Figure 5-17 Stop Transmission Received After Last Data Block. Card is Busy Programming.**



**Figure 5-18 Stop Transmission Received After Last Data Block. Card Becomes Busy.**

**Stream Write**—The data transfer starts  $N_{WR}$  clock cycles after the card response to the sequential write command was received. The bus transaction is identical to that of a write block command (see Figure 5-13). Since the data transfer is not block oriented, the data stream does not include the CRC checksum. Consequently the host cannot receive any CRC status information from the card. The data stream is terminated by a stop command. The bus transaction is identical to the write block option when a data block is interrupted by the stop command (see Figure 5-15).

**Erase, Set and Clear Write Protect Timing**—The host must first tag the sectors to erase using the tag commands (CMD32 - CMD37). The erase command (CMD38), once issued, will erase all tagged sectors. Similarly, set and clear write protect commands start a programming operation as well. The card will signal “busy” (by pulling the DAT line low) for the duration of the erase or programming operation. The bus transaction timings are described in Figure 5-18.

---

#### 5.10.4 Timing Values

Table 5-14 defines all timing values.

**Table 5-14 Timing Values**

	Min	Max	Unit
$N_{CR}$	2	64	Clock Cycles
$N_{ID}$	5	5	Clock Cycles
$N_{AC}$	2	$[10 * ((TAAC*f) + (100*NSAC)) ]^{**}$	Clock Cycles
$N_{RC}$	8	-	Clock Cycles
$N_{CC}$	8	-	Clock Cycles
$N_{WR}$	2	-	Clock Cycles

**\*\* Note:** Where f is the clock frequency

## 6.0 SPI Mode

### 6.1 Introduction

The SPI mode is a secondary, optional communication protocol which is offered by MultiMediaCards. This mode is a subset of the MultiMediaCard protocol, designed to communicate with an SPI channel, commonly found in some vendors' microcontrollers. The interface is selected during the first reset command after power up (CMD0) and cannot be changed once the part is powered on.

The SPI standard defines the physical link only, and not the complete data transfer protocol. MultiMediaCard SPI implementation uses a subset of the MultiMediaCard protocol and command set. It is intended to be used by systems which require a small number of cards (typically one) and have lower data transfer rates (compared to MultiMediaCard protocol based systems). From the application point of view, the advantage of the SPI mode is the capability of using an off-the-shelf host, hence reducing the design-in effort to a minimum. The disadvantage is the loss of performance with SPI mode as compared to MultiMediaCard mode (lower data transfer rate, fewer cards, hardware CS per card, etc.).

### 6.2 SPI Interface Concept

The SPI is a general purpose synchronous serial interface originally found on certain Motorola microcontrollers. A virtually identical interface can now be found on some other microcontrollers as well.

The MultiMediaCard SPI interface is compatible with SPI hosts available on the market. As in any other SPI device, the MultiMediaCard SPI channel consists of the following four signals:

<b>CS:</b>	Host to card Chip Select signal.
<b>CLK:</b>	Host to card clock signal
<b>DataIn:</b>	Host to card data signal.
<b>DataOut:</b>	Card to host data signal.

Byte transfers are another common SPI characteristic. They are implemented in the card as well. All data tokens are multiples of bytes (8 bit) and always byte aligned to the CS signal.

### 6.3 SPI Bus Topology

The card identification and addressing methods are replaced by a hardware Chip Select (CS) signal. There are no broadcast commands. For every command, a card (slave) is selected by asserting (active low) the CS signal. See the following figure.

The CS signal must be continuously active for the duration of the SPI transaction (command, response and data). The only exception occurs during card programming when the host can de-assert the CS signal without affecting the programming process.

The bidirectional CMD and DAT lines are replaced by unidirectional *dataIn* and *dataOut* signals. This eliminates the ability of executing commands while data is being read or written and, therefore, makes the sequential and multi block read/write operations obsolete. Only single block read/write commands are supported by the SPI channel.

The SPI interface uses the same seven signals as the standard MultiMediaCard bus.

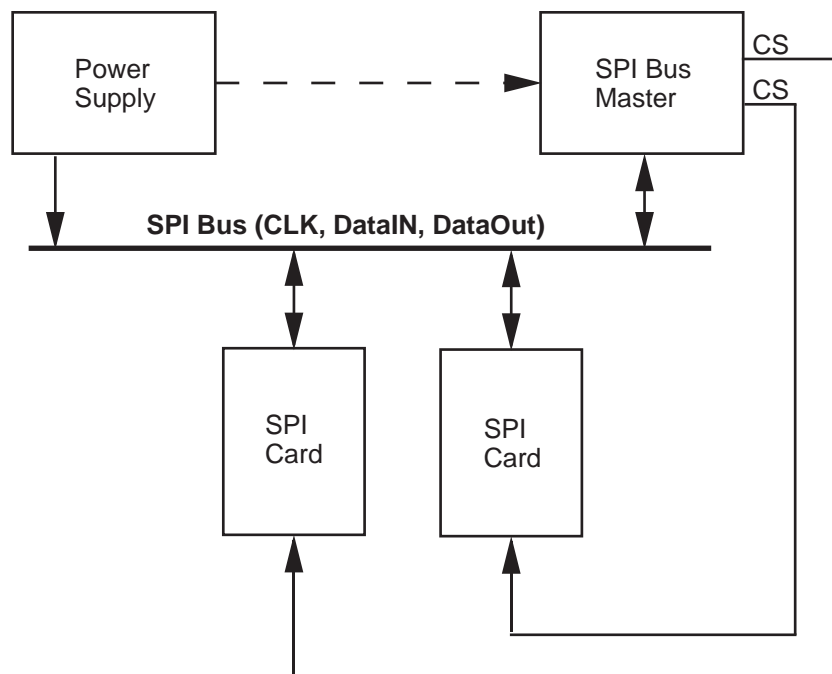


Figure 6-1 MultiMediaCard Bus System

Table 6-1 SPI Interface Pin Configuration

Pin #	MultiMediaCard Mode			SPI Mode		
	Name	Type <sup>1</sup>	Description	Name	Type	Description
1	RSV	NC	Reserved for future use	CS	I	Chip Select (neg true)
2	CMD	I/O/PP/OD	Command/Response	DI	I/PP	Data In
3	VSS1	S	Supply voltage ground	VSS	S	Supply voltage ground
4	VDD	S	Supply voltage	VDD	S	Supply voltage
5	CLK	I	Clock	SCLK	I	Clock
6	VSS2	S	Supply voltage ground	VSS2	S	Supply voltage ground
7	DAT	I/O/PP	Data	DO	O/PP	Data Out

1) S: power supply; I: input; O: output; PP: push-pull; OD: open-drain; NC: Not connected (or logical high).

## 6.4 MultiMediaCard Registers in SPI Mode

The register usage in SPI mode is summarized in the following table. Most of them are inaccessible.

**Table 6-2 MultiMediaCard Registers in SPI Mode**

Name	Available in SPI mode	Width [Bytes]	Description
CID	Yes	16	Card identification data (serial number, manufacturer ID, etc.)
RCA	No		
DSR	No		
CSD	Yes	16	Card-specific data, information about the card operation conditions.
OCR	Yes	32	Operation condition register.

## 6.5 SPI Bus Protocol

While the MultiMediaCard channel is based on command and data bit streams which are initiated by a start bit and terminated by a stop bit, the SPI channel is byte oriented. Every command or data block is built of 8-bit bytes and is byte aligned to the CS signal (i.e. the length is a multiple of 8 clock cycles).

Similar to the MultiMediaCard protocol, SPI messages consist of command, response and data-block tokens. All communication between host and cards is controlled by the host (master). The host starts every bus transaction by asserting the CS signal low.

The response behavior in SPI mode differs from MultiMediaCard mode in the following three aspects:

- The selected card always responds to the command.
- Additional (8, 16 & 40 bit) response structures are used
- When the card encounters a data retrieval problem, it will respond with an error response (which replaces the expected data block) rather than by a time-out, as in the MultiMediaCard mode.

Only single and multiple<sup>1</sup> block read/write operations are supported in SPI mode (sequential mode is not supported). In addition to the command response, every data block sent to the card during write operations will be responded to with a special data response token. A data block may be as big as one card sector and as small as a single byte. Partial block read/write operations are enabled by card options specified in the CSD register.

### 6.5.1 Mode Selection

The MultiMediaCard wakes up in MultiMediaCard mode. It will enter SPI mode if the CS signal is asserted (negative) during the reception of the reset command (CMD0). Selecting SPI mode is not restricted to *Idle* state (the state the card enters after power up) only. Every time the card receives CMD0, including while in *Inactive* state, CS signal is sampled.

<sup>1</sup>) Multiple sector Read/Write in SPI mode was approved by the MMCA and is included in MMCA system standard rev 3.1.

If the card recognizes that MultiMediaCard mode is required (CS signal is high), it will not respond to the command and remain in MultiMediaCard mode. If SPI mode is required (CS signal is low), the card will switch to SPI and respond with the SPI mode R1 response.

The only way to return to MultiMediaCard mode is by a power cycle (turning the power off and on). In SPI mode, the MultiMediaCard protocol state machine is not observed. All of the MultiMediaCard commands supported in SPI mode are always available.

---

### 6.5.2 Bus Transfer Protection

Every MultiMediaCard token transferred on the bus is protected by CRC bits. In SPI mode, the MultiMediaCard offers a non-protected mode which enables systems built with reliable data links to exclude the hardware or firmware required for implementing the CRC generation and verification functions.

In the non-protected mode, the CRC bits of the command, response and data tokens are still required in the tokens. However, they are defined as 'don't care' for the transmitter and ignored by the receiver.

The SPI interface is initialized in the non-protected mode. However, the RESET command (CMD0), which is used to switch the card to SPI mode, is received by the card while in MultiMediaCard mode and, therefore, must have a valid CRC field.

Since CMD0 has no arguments, the content of all the fields, including the CRC field, are constants and need not be calculated in run time. A valid reset command is:

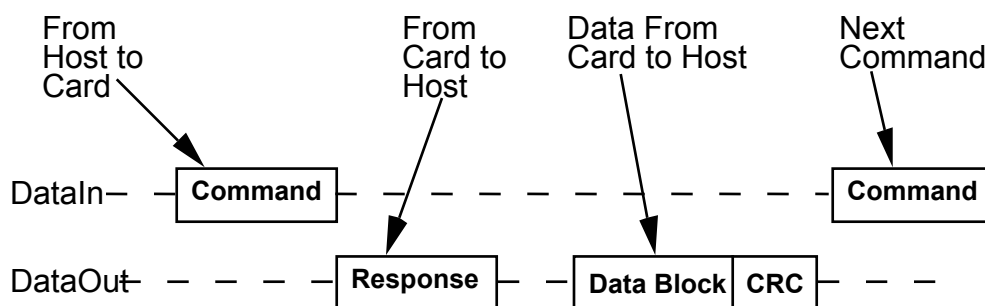
0x40, 0x0, 0x0, 0x0, 0x0, 0x95

The host can turn the CRC option on and off using the CRC\_ON\_OFF command (CMD59).

---

### 6.5.3 Data Read

SPI mode supports single and multiple<sup>1</sup> block read operations. The main difference between SPI and MultiMediaCard modes is that the data and the response are both transmitted to the host on the DataOut signal. Therefore, the card response to the STOP\_COMMAND may cut-short and replace the last data block.

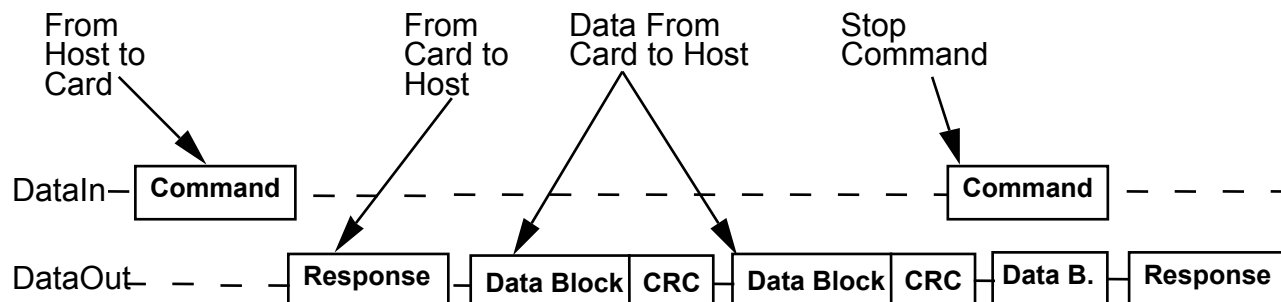


**Figure 6-2 Single Block Read Operation**

---

<sup>1</sup>) Multiple sector Read/Write in SPI mode was approved by the MMCA and is included in MMCA system standard rev 3.1.

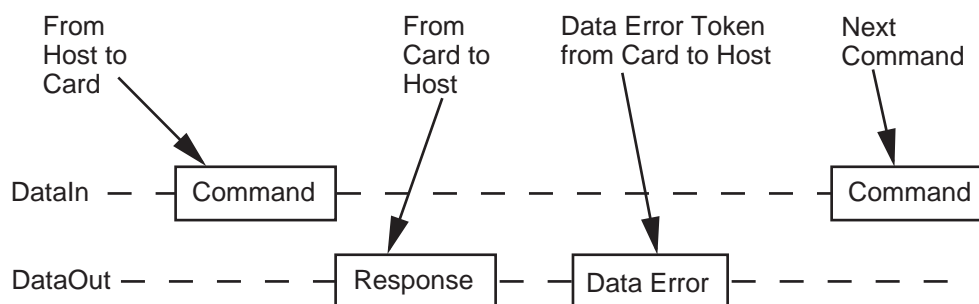




**Figure 6-3 Multiple Block Read Operation**

The basic unit of data transfer is a block whose maximum size is defined in the CSD (READ\_BL\_LEN). If READ\_BL\_PARTIAL is set, smaller blocks whose starting and ending address are entirely contained within one physical block (as defined by READ\_BL\_LEN) may also be transmitted. A CRC is appended to the end of each block ensuring data transfer integrity. CMD17 (READ\_SINGLE\_BLOCK) initiates a single block read. CMD18 (READ\_MULTIPLE\_BLOCK) starts a transfer of several consecutive blocks. The number of blocks for the multiple block read operation is not defined. The card will continuously transfer data blocks until a stop transmission command is received.

In case of a data retrieval error (for example out of range, address misalignment, internal error, etc.), the card will not transmit any data. Instead a special data error token will be sent to the host, as opposed to MultiMediaCard mode where the card times out. The figure below shows a single block read operation which terminates with an error token rather than a data block.



**Figure 6-4 Read Operation - Data Error**

The multiple block read operation can be terminated the same way by the error token replacing a data block anywhere in the sequence. The host must then abort the operation by sending the Stop Transmission command.

If the host sends a Stop Transmission command out of the valid sequence, it will be responded to as an illegal command.

If the host uses partial blocks whose accumulated length is not block aligned and block misalignment is not allowed, the card detects a block misalignment error condition at the beginning of the first misaligned block (ADDRESS\_ERROR error bit is set in the data error token).

## 6.5.4 Data Write

SPI mode supports single block and multiple<sup>1</sup> block write commands. Upon reception of a valid write command (CMD24 or CMD25), the card will respond with a response token and will wait for a data block to be sent from the host. CRC suffix, block length and start address restrictions are (with the exception of the CSD parameter WRITE\_BL\_PARTIAL controlling the partial block write option) identical to the read operation. If a CRC error is detected it will be reported in the data-response token and the data block will not be programmed.

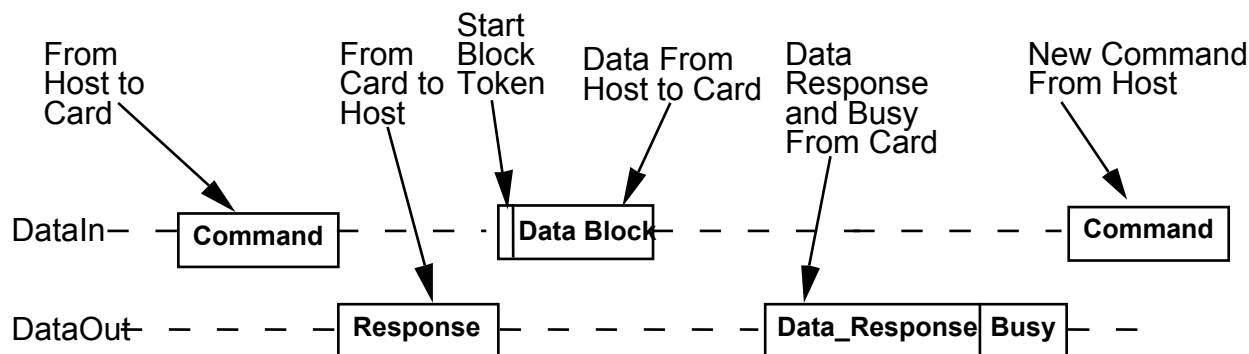


Figure 6-5 Single Block Write Operation

Every data block has a 'Start Block' token prefix (one byte).

After a data block has been received, the card will respond with a data-response token. If the data block has been received without errors, it will be programmed. As long as the card is busy programming, a continuous stream of busy tokens will be sent to the host (effectively holding the DataOut line low).

In the Multiple Block Write operation the Stop Transmission will be done by sending a 'Stop Tran' token instead of 'Start Block' token at the beginning of the next block.

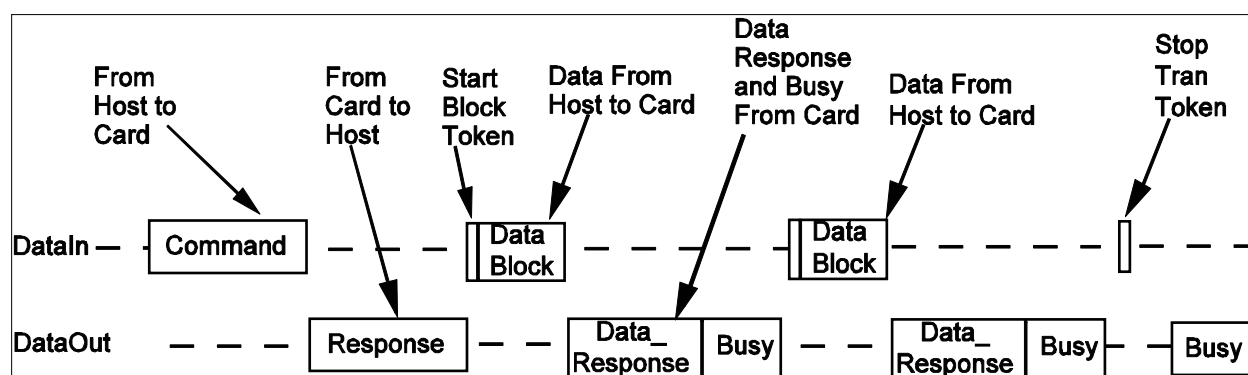


Figure 6-6 Multiple Block Write Operation

The number of blocks for the write multiple block operation is not defined. The card will continuously accept and program data blocks until a 'Stop Tran' token is received.

<sup>1</sup>) Multiple sector Read/Write in SPI mode was approved by the MMCA and is included in MMCA system standard rev 3.1.

If the card detects a CRC error or a programming error (for example, write protect violation, out of range, address misalignment, internal error, etc.) during a multiple block write operation, it will report the failure in the data-response token and ignore any further incoming data blocks. The host must then abort the operation by sending the 'Stop Tran' token.

If the host uses partial blocks whose accumulated length is not block aligned and block misalignment is not allowed (CSD parameter WRITE\_BLK\_MISALIGN is not set), the card detects the block misalignment error before the beginning of the first misaligned block and responds with an error indication in the data response token, ignoring any further incoming data blocks. The host must then abort the operation by sending the 'Stop Tran' token.

Once the programming operation is completed (either successfully or with an error), the host must check the results of the programming (or the cause of the error if already reported in the data-response token) using the SEND\_STATUS command (CMD13).

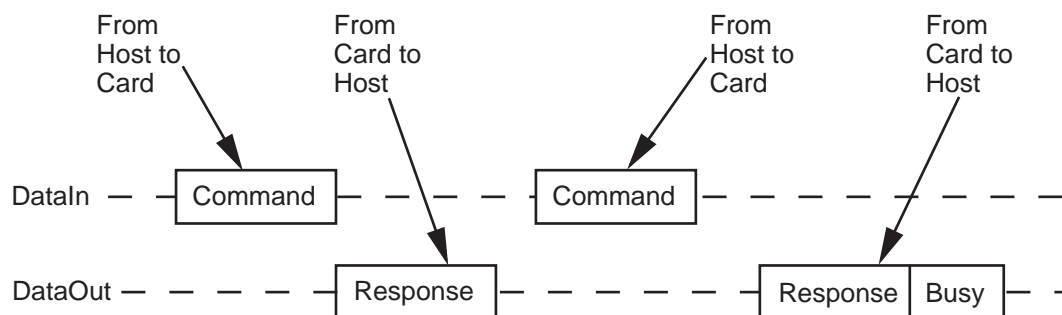
While the card is busy, resetting the CS signal will not terminate the programming process. The card will release the DataOut line (tri-state) and continue with programming. If the card is reselected before the programming is finished, the DataOut line will be forced back to low and all commands will be rejected.

Resetting a card (using CMD0) will terminate any pending or active programming operations. This may destroy the data formats on the card. It is the responsibility of the host to prevent it.

---

#### 6.5.5 Erase & Write Protect Management

The erase and write protect management procedures in SPI mode are identical to those in MultiMediaCard mode. While the card is erasing or changing the write protection bits of the predefined erase groups list, it will be in a busy state and hold the DataOut line low. The figure below illustrates a 'no data' bus transaction with and without busy signaling.



**Figure 6-7 'No data' Operations**

---

#### 6.5.6 Read CID/CSD Registers

Unlike the MultiMediaCard protocol, where the register content is sent as a command response, reading the contents of the CSD and CID registers in SPI mode is a simple read-block transaction. The card will respond with a standard response token followed by a data block of 16 bytes suffixed with a 16 bit CRC.

The data time out for the CSD command cannot be sent to the card TAAC since this value is stored in the CSD. Therefore, the standard response time-out value (N<sub>CR</sub>) is used for read latency of the CSD register.

---

#### 6.5.7 Reset Sequence

The MultiMediaCard requires a defined reset sequence. After power on reset or CMD0 (software reset), the card enters an idle state. In this state, the only legal host commands are CMD1 (SEND\_OP\_COND) and CMD58 (READ\_OCR).

The host must poll the card (by repeatedly sending CMD1) until the 'in-idle-state' bit in the card response indicates (by being set to 0) that the card has completed its initialization processes and is ready for the next command.

In SPI mode, as opposed to MultiMediaCard mode, CMD1 has no operands and does not return the contents of the OCR register. Instead, the host may use CMD58 (available in SPI mode only) to read the OCR register. Furthermore, it is the host's responsibility to refrain from accessing cards that do not support its voltage range.

The usage of CMD58 is not restricted to the initializing phase only, but can be issued at any time. The host must poll the card (by repeatedly sending CMD1) until the 'in-idle-state' bit in the card response indicates (by being set to 0) that the card has completed its initialization processes and is ready for the next command.

---

#### 6.5.8 Clock Control

The SPI bus clock signal can be used by the SPI host to set the cards to energy saving mode or to control the data flow (to avoid under-run or over-run conditions) on the bus. The host is allowed to change the clock frequency or shut it down.

There are a few restrictions the SPI host must follow:

- The bus frequency can be changed at any time under the restrictions of maximum data transfer frequency, defined by the MultiMediaCards.
- The clock must be running for the MultiMediaCard to output data or response tokens. After the last SPI bus transaction, the host is required to provide eight clock cycles for the card to complete the operation before shutting down the clock. Throughout this eight clock period, the state of the CS signal is irrelevant. It can be asserted or de-asserted. Following is a list of the various SPI bus transactions:
  - A command / response sequence - eight clocks after the card response end bit. The CS signal can be asserted or de-asserted during these eight clock cycles.
  - A read data transaction - eight clocks after the end bit of the last data block.
  - A write data transaction - eight clocks after the CRC status token.

- The host is allowed to shut down the clock of a busy card. The MultiMediaCard will complete the programming operation regardless of the host clock. However, the host must provide a clock edge for the card to turn off its busy signal. Without a clock edge, the MultiMediaCard (unless previously disconnected by de-asserting the CS signal) will force the dataOut line down, permanently.

---

#### 6.5.9 Error Conditions

---

##### 6.5.9.1 CRC and Illegal Command

All commands are optionally protected by CRC (cyclic redundancy check) bits. If the addressed MultiMediaCard's CRC check fails, the COM\_CRC\_ERROR bit will be set in the card's response. Similarly, if an illegal command has been received, the ILLEGAL\_COMMAND bit will be set in the card's response.

There are different kinds of illegal commands:

- Commands which belong to classes not supported by the MultiMediaCard (such as interrupt and I/O commands).
- Commands not allowed in SPI mode (for example, CMD20 - write stream).
- Commands which are not defined (for example, CMD6).

---

##### 6.5.9.2 Read, Write and Erase Time-out Conditions

The time period after which a time-out condition for read/write/erase operations occurs is ten times longer than the typical access/program times for these operations given below (card independent). A card shall complete the command within this time period, or give up and return an error message. If the host does not get a response within the defined time-out, it should assume the card is not going to respond any more and try to recover (that is, reset the card, power cycle, reject, etc.).

The typical access and program times are defined as follows:

- Read—The read access time is defined as the sum of the two times given by the CSD parameters TAAC and NSAC. These card parameters define the typical delay between the end bit of the read command and the start bit of the data block. This number is card dependent.
- Write—The R2W\_FACTOR field in the CSD is used to calculate the typical block program time obtained by multiplying the read access time by this factor. It applies to all write/erase commands (such as SET(CLEAR)\_WRITE\_PROTECT, PROGRAM\_CSD(CID) and block write commands).
- Erase—The duration of an erase command will be (order of magnitude) the number of sectors to be erased multiplied by the block write delay.

---

#### 6.5.9.3 Read Ahead in Multiple Block Read Operation

In Multiple Block read operations, in order to improve read performance, the card may fetch data from the memory array, ahead of the host. In this case, when the host is reading the last addresses of the memory, the card attempts to fetch data beyond the last physical memory address and generates an OUT\_OF\_RANGE error. Therefore, even if the host times the Stop Transmission command to stop the card immediately after the last byte of data was read, the card may already have generated the error, which will show in the response to the Stop Transmission command. The host should ignore this error.

---

#### 6.5.10 Memory Array Partitioning

Same as for MultiMediaCard mode.

## 6.6 SPI Mode Transaction Packets

### 6.6.1 Command Tokens

#### 6.6.1.1 Command Format

All the MultiMediaCard commands are six bytes long. The command transmission always starts with the left bit of the bit string corresponding to the command code word. All commands are protected by a CRC. The commands and arguments are listed in Table 6-5.

**Table 6-3 Command Format**

Bit Position	47	46	[45:40]	[39:8]	[7:1]	0
Width (bits)	1	1	6	32	7	1
Value	'0'	'1'	x	x	x	011
Description	start bit	transmission bit	command index	argument	CRC7	end bit

#### 6.6.1.2 Command Classes

As in MultiMediaCard mode, the SPI commands are divided into several classes. (See the table below.) Each class supports a set of card functions. A MultiMediaCard will support the same set of optional command classes in both communication modes. (There is only one command class table in the CSD register.) The available command classes, and the supported commands for a specific class, however, are different in the MultiMediaCard and the SPI communication modes.

**Table 6-4 Command Classes in SPI Mode**

Card CMD Class (CCC)	Class Description	Supported Commands																											
		0	1	9	10	12	13	16	17	18	23	24	25	27	28	29	30	35	36	38	42	55	56	58	59				
class 0	Basic	+	+	+	+		+																		+	+			
class 1	Not supported in SPI																												
class 2	Block read					+		+	+	+	+																		
class 3	Not supported in SPI																												
class 4	Block write							+			+	+	+	+															
class 5	Erase																		+	+	+								
class 6 <sup>1</sup>	Write-protection															+	+	+											
class 7	Lock Card																				+								
class 8	Application specific																					+	+						
class 9	Not supported in SPI																												
class 10-11	Reserved																												

<sup>1</sup>) Group write protect, Application specific, card lock and I/O command classes are not supported

## 6.6.1.3 Detailed Command Description

The following table provides a detailed description of the SPI mode commands. The responses are defined in section 6.6.2. The table below lists all MultiMediaCard commands. A 'yes' in the SPI mode column indicates that the command is supported in SPI mode. With these restrictions, the command class description in the CSD is still valid. If a command does not require an argument, the value of this field should be set to zero. The reserved SPI mode commands are also reserved in MultiMediaCard mode.

The binary code of a command is defined by the mnemonic symbol. As an example, the content of the **command index** field is (binary) '000000' for CMD0 and '100111' for CMD39.

**Table 6-5 Commands and Arguments**

CMD INDEX	SPI Mode	Argument	Resp	Abbreviation	Command Description
CMD0	Yes	None	R1	GO_IDLE_STATE	Resets the MultiMediaCard.
CMD1	Yes	None	R1	SEND_OP_COND	Activates the card's initialization process.
CMD2	No				
CMD3	No				
CMD4	No				
CMD5	reserved				
CMD6	reserved				
CMD7	No				
CMD8	reserved				
CMD9	Yes	None	R1	SEND_CSD	Asks the selected card to send its card-specific data (CSD).
CMD10	Yes	None	R1	SEND_CID	Asks the selected card to send its card identification (CID).
CMD11	No				
CMD12	Yes	None	R1	STOP_TRANSMISSION	Stop transmission on multiple block read.
CMD13	Yes	None	R2	SEND_STATUS	Asks the selected card to send its status register.
CMD14	reserved				
CMD15	No				
CMD16	Yes	[31:0] block length	R1	SET_BLOCKLEN	Selects a block length (in bytes) for all following block commands (read and write). <sup>1</sup>
CMD17	Yes	[31:0] data address	R1	READ_SINGLE_BLOCK	Reads a block of the size selected by the SET_BLOCKLEN command. <sup>2</sup>
CMD18	Yes	[31:0] data address	R1	READ_MULTIPLE_BLOCK	Continuously transfers data blocks from card to host until interrupted by a Stop command or the requested number of data blocks transmitted.

1) The default block length is as specified in the CSD.

2) The data transferred must not cross a physical block boundary unless READ\_BLK\_MISALIGN is set in the CSD.



CMD19	reserved				
CMD20	No				
CMD21 ... CMD23	reserved				
CMD24	Yes	[31:0] data address	R1	WRITE_BLOCK	Writes a block of the size selected by the SET_BLOCKLEN command. <sup>1</sup>
CMD25	Yes	[31:0] data address	R1	WRITE_MULTIPLE_BLOCK	Continuously writes blocks of data until a 'Stop Tran' Token or the requested number of blocks received.
CMD26	No				
CMD27	Yes	None	R1	PROGRAM_CSD	Programming of the programmable bits of the CSD.
CMD28	Yes	[31:0] data address	R1b <sup>2</sup>	SET_WRITE_PROT	If the card has write protection features, this command sets the write protection bit of the addressed group. The properties of write protection are coded in the card specific data (WP_GRP_SIZE).
CMD29	Yes	[31:0] data address	R1b	CLR_WRITE_PROT	If the card has write protection features, this command clears the write protection bit of the addressed group.
CMD30	Yes	[31:0] write protect data address	R1	SEND_WRITE_PROT	If the card has write protection features, this command asks the card to send the status of the write protection bits. <sup>3</sup>
CMD31	reserved				
CMD32	Yes	[31:0] data address	R1	TAG_SECTOR_START	Sets the address of the first sector of the erase group.
CMD33	Yes	[31:0] data address	R1	TAG_SECTOR_END	Sets the address of the last sector in a continuous range within the selected erase group, or the address of a single sector to be selected for erase.
CMD34	Yes	[31:0] data address	R1	UNTAG_SECTOR	Removes one previously selected sector from the erase selection.
CMD35	Yes	[31:0] data address	R1	TAG_ERASE_GROUP_START	Sets the address of the first erase group within a range to be selected for erase.
CMD36	Yes	[31:0] data address	R1	TAG_ERASE_GROUP_END	Sets the address of the last erase group within a continuous range to be selected for erase.
CMD37	Yes	[31:0] data address	R1	UNTAG_ERASE_GROUP	Removes one previously selected erase group from the erase selection.
CMD38	Yes	[31:0] stuff bits	R1b	ERASE	Erases all previously selected sectors.

1) The data transferred must not cross a physical block boundary unless WRITE\_BLK\_MISALIGN is set in the CSD.

2) R1b: R1 response with an optional trailing busy signal.

3) 32 write protection bits (representing 32 write protect groups starting at the specified address) followed by 16 CRC bits are transferred in a payload format via the data line. The last (least significant) bit of the protection bits corresponds to the first addressed group. If the addresses of the last groups are outside the valid range, then the corresponding write protection bits are set to zero.

CMD39	No				
CMD40	No				
CMD41	reserved				
CMD42	Yes	This optional MMCA command is not supported in the SanDisk MultiMediaCard			
CMD43 ... CMD54	reserved				
CMD55	Yes	This optional MMCA command is not supported in the SanDisk MultiMediaCard			
CMD56	Yes	This optional MMCA command is not supported in the SanDisk MultiMediaCard			
CMD57	Reserved				
CMD58	Yes	None	R3	READ_OCR	Reads the OCR register of a card.
CMD59	Yes	[31:1] stuff bits [0:0] CRC option	R1	CRC_ON_OFF	Turns the CRC option on or off. A ‘1’ in the CRC option bit will turn the option on, a ‘0’ will turn it off.
CMD60... CMD63	No				

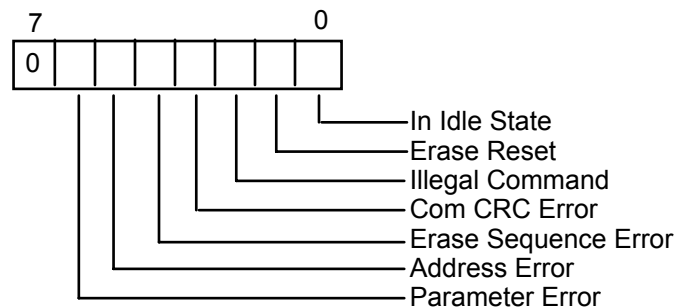
### 6.6.2 Responses

There are several types of response tokens. As with MultiMediaCard mode, all are transmitted MSB first.

### 6.6.2.1 Format R1

This response token is sent by the card after every command, with the exception of SEND\_STATUS commands. It is one byte long, and the MSB is always set to zero. The other bits are error indications, an error being signaled by a '1.' The structure of the R1 format is given in the figure below. The meaning of the flags is defined as follows:

- **Idle state:** The card is in idle state and running the initializing process.
- **Erase reset:** An erase sequence was cleared before executing because an out of erase sequence command was received.
- **Illegal command:** An illegal command code was detected.
- **Communication CRC error:** The CRC check of the last command failed.
- **Erase sequence error:** An error occurred in the sequence of erase commands.
- **Address error:** A misaligned address, which did not match the block length, was used in the command.
- **Parameter error:** The command's argument (for example, address, block length) was out of the allowed range for this card.



### Figure 6-8 R1 Response Format

#### 6.6.2.2 Format R1b

This response token is identical to the R1 format with the optional addition of the busy signal.

### 6.6.2.3 Busy

The busy signal token can be any number of bytes. A zero value indicates that the card is busy. A non-zero value indicates that the card is ready for the next command.

## 6.6.2.4 Format R2

This response token is two bytes long and sent as a response to the SEND\_STATUS command. The format is given in the following figure.

The first byte is identical to the response R1. The content of the second byte is described as follows:

- **Out of range|csd\_overwrite**—This status bit has two functions. It is set if the command argument was out of its valid range, or if the host is trying to change the ROM section or reverse the copy bit (set as original) or permanent WP bit (un-protect) of the CSD register.
- **Erase param**—An invalid selection, sectors or groups, for erase.
- **Write protect violation**—The command tried to write a write-protected block.
- **Card ECC failed**—Card internal ECC was applied but failed to correct the data.
- **CC error**—Internal card controller error.
- **Error**—A general or an unknown error occurred during the operation.
- **Write protect erase skip|lock/unlock command failed**—This status bit has two functions. It is set when the host attempts to erase a write-protected sector or if a sequence or password error occurred during a card lock/unlock operation.
- **Card is locked**—This bit is set when the card is locked by the user. It is reset when it is unlocked.

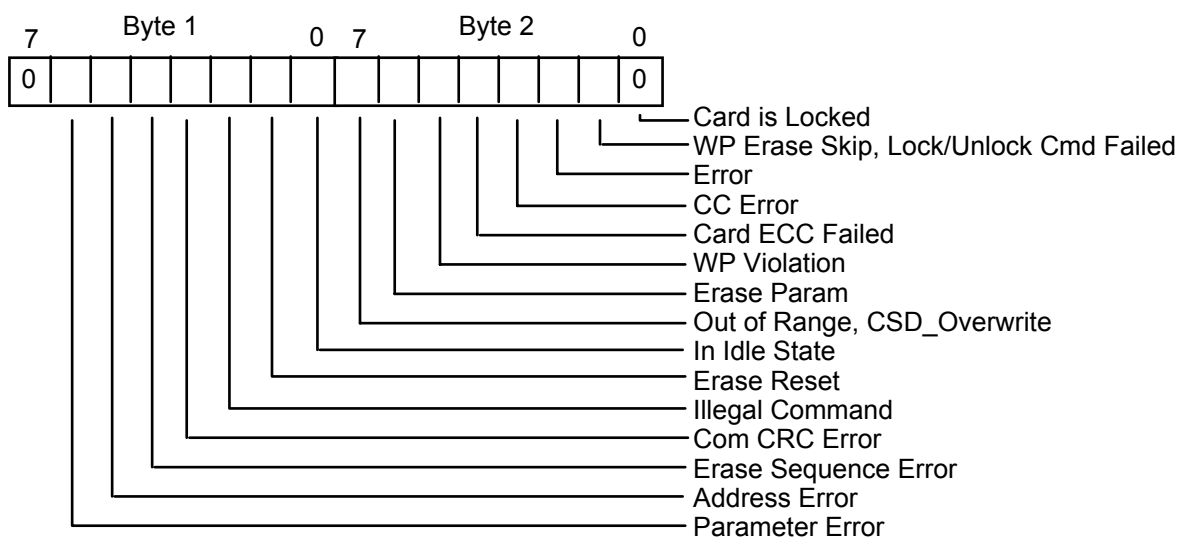
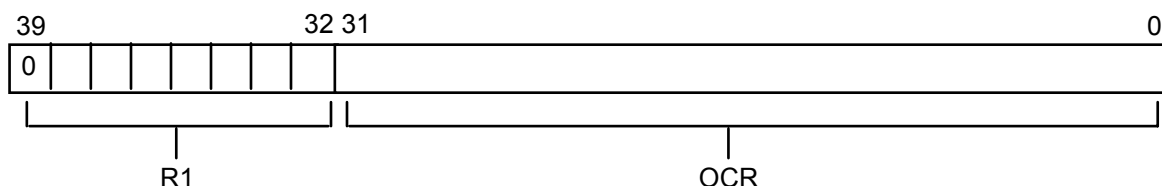


Figure 6-9 R2 Response Format

---

6.6.2.5 Format R3

This response token is sent by the card when a READ\_OCR command is received. The response length is 5 bytes. The structure of the first (MSB) byte is identical to response type R1. The other four bytes contain the OCR register.

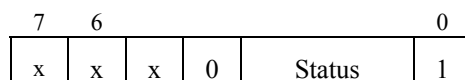


**Figure 6-10 R3 Response Format**

---

6.6.2.6 Data Response

Every data block written to the card will be acknowledged by a data response token. It is one byte long and has the following format:



**Figure 6-11 Data Response Format**

The status bits are defined as follows:

- ‘010’ - Data accepted.
- ‘101’ - Data rejected due to a CRC error.
- ‘110’ - Data rejected due to a Write error.

In case of any error (CRC or Write Error) during a Write Multiple Block operation, the host aborts the operation using the ‘Stop Tran’ token. In case of a Write Error (response ‘110’) the host should send CMD13 (SEND\_STATUS) in order to get the cause of the write problem.

### 6.6.3 Data Tokens

Read and write commands have data transfers associated with them. Data is being transmitted or received via data tokens. All data bytes are transmitted MSB first.

Data tokens are 4 to (N + 3) bytes long (where N is the data block length set using the SET\_BLOCK\_LENGTH command) and have the following format:

- First byte:

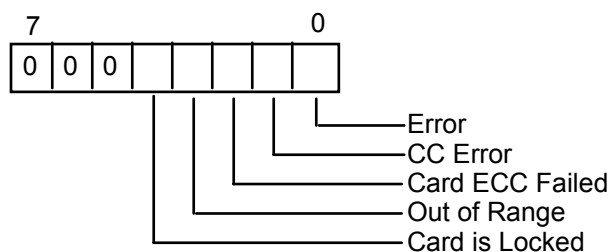
### Table 6-6 Format of the Start Data Block Token

Token Type	Transaction Type	7	6	5	4	3	2	1	0
Start Block	Single Block Read	1	1	1	1	1	1	1	0
Start Block	Multiple Block Read	1	1	1	1	1	1	1	0
Start Block	Single Block Write	1	1	1	1	1	1	1	0
Start Block	Multiple Block Write	1	1	1	1	1	1	0	0
Stop Tran	Multiple Block Write	1	1	1	1	1	1	0	1

- Bytes 2 - (N + 1): User data
- Last two bytes: 16 bit CRC.

#### 6.6.4 Data Error Token

If a read operation fails and the card cannot provide the required data, it will send a data error token instead. This token is one byte long and has the following format:



### Figure 6-12 Data Error Token

The four least significant bits (LSB) are the same error bits as in the response format R2.

### 6.6.5 Clearing Status Bits

As described in the previous paragraphs, in SPI mode, status bits are reported to the host in three different formats: response R1, response R2 and data error token (the same bits may exist in multiple response types, such as Card ECC failed).

As in MultiMediaCard mode, error bits are cleared when read by the host, regardless of the response format. State indicators are either cleared by reading or in accordance with the card state.

The following table summarizes the set and clear conditions for the various status bits:

**Table 6-7 SPI Mode Status Bits**

Identifier	Included in resp.	Type <sup>1</sup>	Value	Description	Clear Cond <sup>2</sup>
Out of range	R2 DataErr	E R X	0 = no error 1 = error	The command argument was out of the allowed range for this card.	C
Address error	R1 R2	E R X	0 = no error 1 = error	An address which did not match the block length was used in the command.	C
Erase sequence error	R1 R2	E R	0 = no error 1 = error	An error in the sequence of erase commands occurred.	C
Erase param	R2	E X	0 = no error 1 = error	An error in the parameters of the erase command sequence.	C
Parameter error	R1 R2	E R X	0 = no error 1 = error	An error in the parameters of the command.	C
WP violation	R2	E R X	0 = not protected 1 = protected	Attempt to program a write protected block.	C
Com CRC error	R1 R2	E R	0 = no error 1 = error	The CRC check of the previous command failed.	C
Illegal command	R1 R2	E R	0 = no error 1 = error	Command not legal for the card state.	C
Card ECC failed	R2 DataEr	E X	0 = success 1 = failure	Card internal ECC was applied but failed to correct the data.	C
CC error	R2 dataEr	E R X	0 = no error 1 = error	Internal card controller error.	C
Error	R2 dataEr	E R X	0 = no error 1 = error	A general or an unknown error occurred during the operation.	C
WP erase skip	R2	S X	0 = not protected 1 = protected	Only partial address space was erased due to existing write protected blocks.	C
Lock/Unlock cmd failed	R2	E X	0 = no error 1 = error	Sequence or password error during card lock/unlock operation.	C
Card is locked	R2 dataEr	S X	0 = card is not locked 1 = card is locked	Card is locked by a user password.	A
Erase reset	R1 R2	S R	0 = cleared 1 = set	An erase sequence was cleared before executing because an out of erase sequence command was received.	C

- 1) **Type:** E = error bit; S = state bit; R = detected and set for the actual response; X = detected and set during command execution (the host must poll the card by issuing the status command in order to read these bits).
- 2) **Clear Condition:** A = according to the card current state; C = clear by read.

In Idle state	R1 R2	S R	0 = Card is ready 1 = Card is in idle state	The card enters the idle state after a power up or reset command. It will exit this state and become ready upon completion of its initialization procedures.	A
CSD overwrite	R2	E X	0 = no error 1 = error	The host is trying to change the ROM section, or is trying to reverse the copy bit (set as original) or permanent WP bit (un-protect) of the CSD register.	C

---

## 6.7 Card Registers

In SPI mode, only the OCR, CSD and CID registers are accessible. Their format is identical to the format in the MultiMediaCard mode. However, a few fields are irrelevant in SPI mode.

---

## 6.8 SPI Bus Timing Diagrams

All timing diagrams use the following schematics and abbreviations.

**Table 6-8 Timing Diagram Abbreviations**

<b>H</b>	Signal is high (logical '1')
<b>L</b>	Signal is low (logical '0')
<b>X</b>	Don't care (Undefined Value)
<b>Z</b>	High impedance state (-> = 1)
<b>*</b>	Repeater
<b>Busy</b>	Busy Token
<b>Command</b>	Command token
<b>Response</b>	Response token
<b>Data Block</b>	Data token

All timing values are defined in Table 6-9. The host must keep the clock running for at least  $N_{CR}$  clock cycles after receiving the card response. This restriction applies to both command and data response tokens.



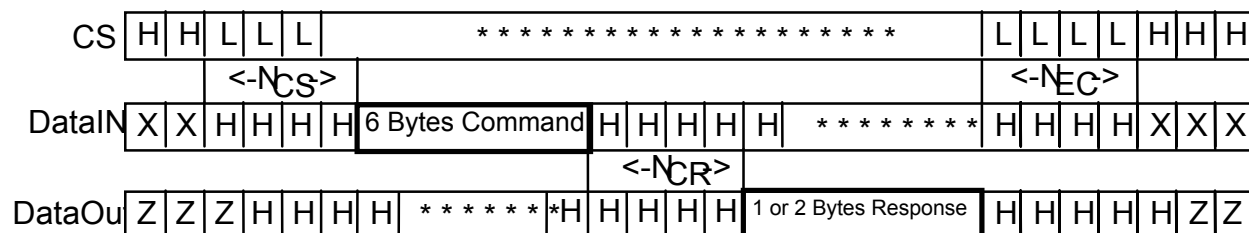
---

6.8.1 Command/Response

---

6.8.1.1 Host Command to Card Response - Card is Ready

The following timing diagram describes the basic command response (no data) SPI transaction.

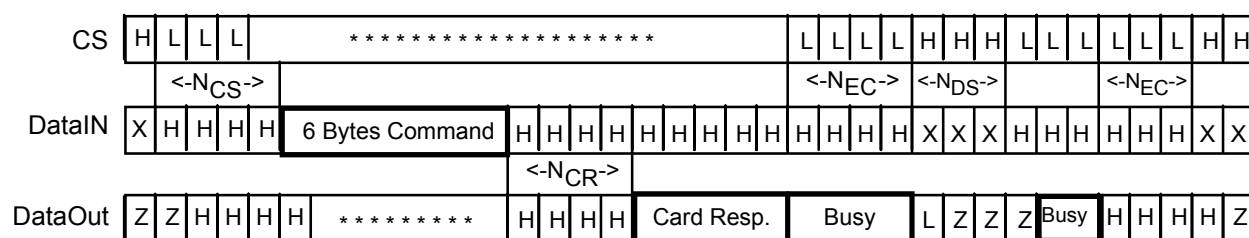


**Figure 6-13 Timing Diagram of Command/Response Transaction, Card is Ready**

---

6.8.1.2 Host Command to Card Response - Card is Busy

The following timing diagram describes the command response transaction for commands when the card response is of type R1b (for example SET\_WRITE\_PROT and ERASE). When the card is signaling busy, the host may de-select it by raising the CS at any time. The card will release the DataOut line one clock after the CS going high. To check if the card is still busy, it needs to be re-selected by asserting the CS signal (set to low). The card will resume the busy signal (pulling DataOut low) one clock after the falling edge of CS.



**Figure 6-14 Timing Diagram of Command/Response Transaction, Card is Busy**

## 6.8.1.3 Card Response to Host Command

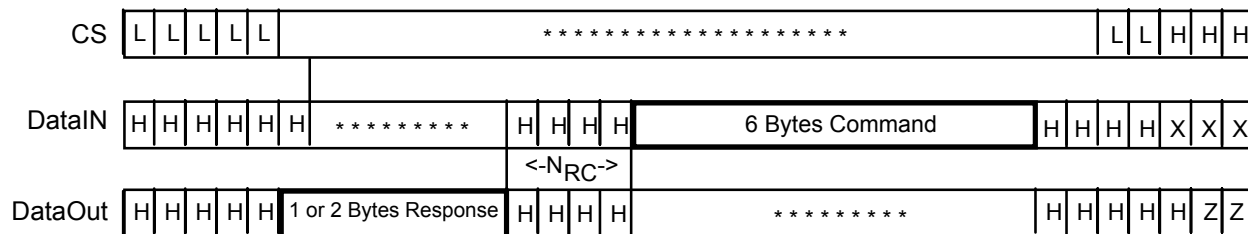


Figure 6-15 Timing Diagram: Card Response to the Next Host Command

## 6.8.2 Data Read

## 6.8.2.1 Single Block Read

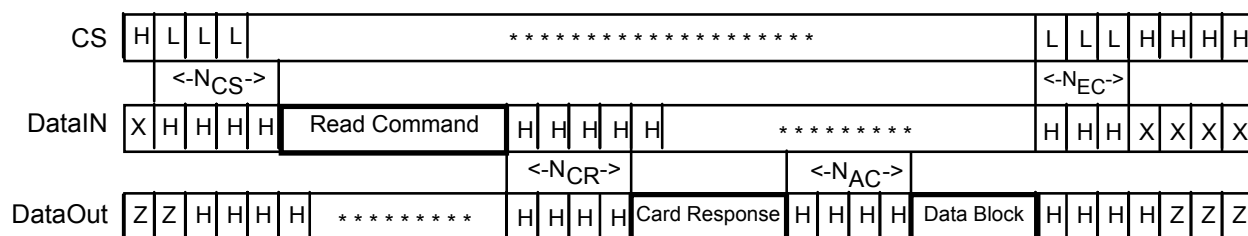
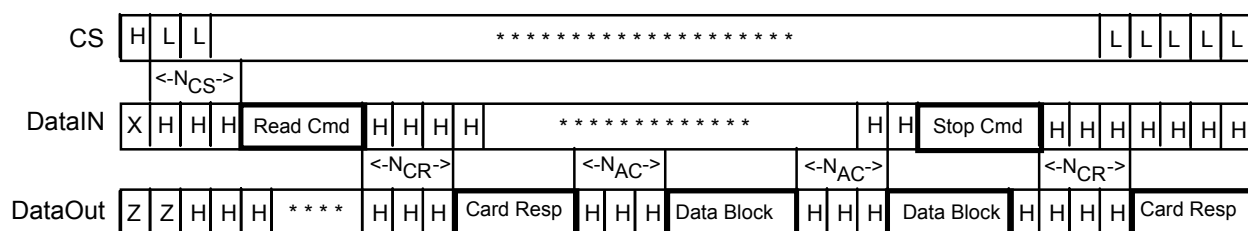


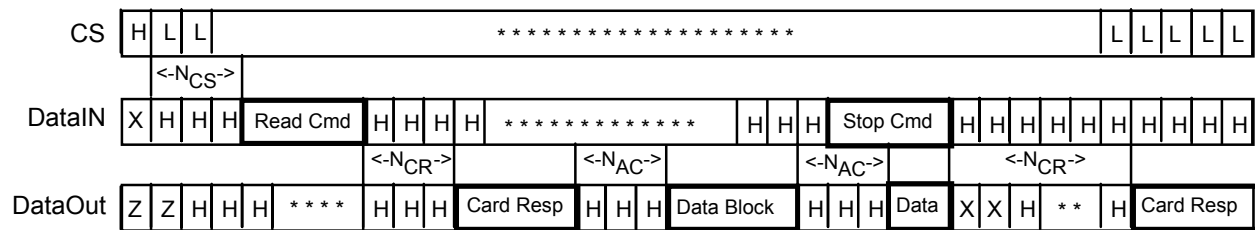
Figure 6-16 Timing Diagram: Single Block Read Transaction

## 6.8.2.2 Multiple Block Read- Stop Transmission is Sent Between Blocks

Figure 6-17 Timing Diagram: Multiple Block Transaction,  
Stop Transmission Does Not Overlap Data

The timing for de-asserting the CS signal after the last card response is identical to a standard command/response transaction as described in Figure 6-13.

### 6.8.2.3 Multiple Block Read- Stop Transmission is Sent Within a Block



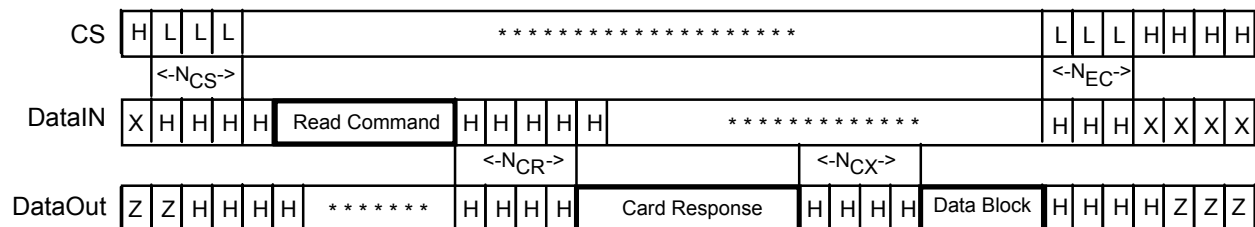
**Figure 6-18 Timing diagram: Multiple Block Transaction, Stop Transmission Overlaps Data**

The Stop Transmission command may be sent asynchronously to the data transmitted out of the card and may overlap the data block. In this case the card will stop sending the data and transmit the response token as well. The delay between command and response is standard  $N_{CR}$  Clocks. The first byte, however, is not guaranteed to be all set to '1.' The card is allowed up to two clocks to stop data transmission.

The timing for de-asserting the CS signal after the last card response is identical to a standard command/response transaction as described in Figure 6-13.

#### 6.8.2.4 Reading the CSD Register

The following timing diagram describes the SEND\_CSD command bus transaction. The time-out values between the response and the data block is  $N_{CX}$ , since the  $N_{AC}$  is still unknown.

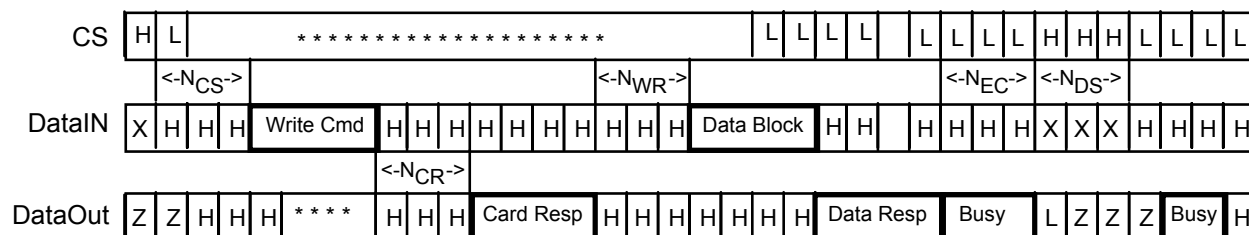


### Figure 6-19 Timing diagram: Read CSD Register

### 6.8.3 Data Write

### 6.8.3.1 Single Block Write

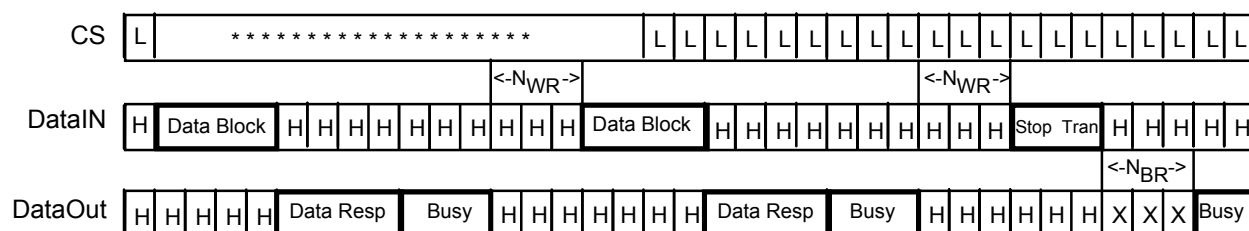
The host may de-select a card by raising the CS at any time during the card busy period. (Refer to the given timing diagram.) The card will release the DataOut line one clock after the CS going high. To check if the card is still busy, it needs to be re-selected by asserting the CS signal (set to low). The card will resume the busy signal (pulling DataOut low) one clock cycle after the falling edge of CS.



### Figure 6-20 Timing Diagram: Single Block Write

### 6.8.3.2 Multiple Block Write

The timing of the multiple block write transaction starting from the command up to the first data block is identical to the single block write. The following figure describes the timing between the data blocks of a multiple block write transaction. Timing of the 'Stop Tran' token is identical to a standard data block. After the 'Stop Tran' token is received by the card, the data on the DataOut line is undefined for one byte ( $N_{BR}$ ), after which a Busy token may appear. The host may de-select and re-select the card during every busy period between the data blocks. Timing for toggling the CS signal is identical to the Single block write transaction



### Figure 6-21 Timing Diagram: Multiple Block Write

#### 6.8.4 Timing Values

### Table 6-9 Timing Values

	Min	Max	Unit
N <sub>CS</sub>	0	-	8 clock cycles
N <sub>CR</sub>	1	8	8 clock cycles
N <sub>CX</sub>	0	8	8 clock cycles
N <sub>RC</sub>	1	-	8 clock cycles
N <sub>AC</sub>	1	$[10*((TAAC*f)+(100*NSAC))]*1/8$ **	8 clock cycles
N <sub>WR</sub>	1	-	8 clock cycles
N <sub>EC</sub>	0	-	8 clock cycles
N <sub>DS</sub>	0	-	8 clock cycles
N <sub>RR</sub>	1	1	8 clock cycles

\*\* Note: Where f is the clock frequency

---

## 6.9 SPI Electrical Interface

Identical to MultiMediaCard mode with the exception of the programmable card output drivers option, which is not supported in SPI mode.

---

## 6.10 SPI Bus Operating Conditions

Identical to MultiMediaCard mode.

---

## 6.11 Bus Timing

Identical to MultiMediaCard mode. The timing of the CS signal is the same as any other card input.

*This page intentionally left blank*

# Ordering Information and Technical Support

*This page intentionally left blank*



## Ordering Information

To order SanDisk products directly from SanDisk, call 408-542-0595.

---

### MultiMediaCard

Model	SDMB-8	8 MB
	SDMB-16	16 MB
	SDMB-32	32 MB
	SDMB-48	48 MB
	SDMB-64	64 MB

## Technical Support Services

---

### Direct SanDisk Technical Support

Call SanDisk Applications Engineering at 408-542-0405 for technical support.

---

### SanDisk Worldwide Web Site

Internet users can obtain technical support and product information along with SanDisk news and much more from the SanDisk Worldwide Web Site, 24 hours a day, seven days a week. The SanDisk Worldwide Web Site is frequently updated. Visit this site often to obtain the most up-to-date information on SanDisk products and applications. The SanDisk Web Site URL is <http://www.sandisk.com>.

## SanDisk Sales Offices

*This page intentionally left blank*

## SanDisk Worldwide Sales Offices

### **Americas**

#### **SanDisk Corporate Headquarters**

140 Caspian Court  
Sunnyvale, CA 94089-9820  
408-542-0500  
FAX 408-542-0503  
<http://www.sandisk.com>

### **Sales Offices**

#### **Western Region USA**

408-542-0730  
FAX 408-542-0403

#### **Eastern Region USA & Canada**

603-882-0888  
FAX 603-882-2207

#### **Central & Southern Region USA**

614-760-3700  
FAX 614-760-3701

#### **Latin & South America**

407-667-4880  
FAX 407-667-4834

### **Europe**

#### **SanDisk GmbH**

Karlsruher Str. 2C  
D-30519 Hannover, Germany  
49-511-8759185  
FAX 49-511-8759187

#### **SanDisk Northern Europe**

Videroegaten 3 B  
S-16440 Kista  
Sweden  
46-(0)8-75084-63  
FAX 46-(0)8-75084-26

#### **SanDisk Central Europe**

Eutelis Plaz 3  
D-40878 Ratingen  
Germany  
49-2102-999666  
FAX 49-2102-999667

### **Japan**

#### **SanDisk K.K.**

8F Nisso Bldg. 15  
2-17-19 Shin-Yokohama, Kohoku-ku  
Yokohama 222-0033, Japan  
81-45-474-0181  
FAX 81-45-474-0371

### **Asia/Pacific Rim**

89 Queensway, Lippo Center  
Tower II, Suite 4104  
Admiralty, Hong Kong  
852-2712-0501  
FAX 852-2712-9385

To order SanDisk products directly from SanDisk,  
call 408-542-0595.

*This page intentionally left blank*

## Limited Warranty

### I. WARRANTY STATEMENT

SanDisk warrants its products to be free of any defects in materials or workmanship that would prevent them from functioning properly for one year from the date of purchase. This express warranty is extended by SanDisk Corporation.

### II. GENERAL PROVISIONS

This warranty sets forth the full extent of SanDisk's responsibilities regarding the SanDisk MultiMediaCard. In satisfaction of its obligations hereunder, SanDisk, at its sole option, will either repair, replace or refund the purchase price of the product.

NOTWITHSTANDING ANYTHING ELSE IN THIS LIMITED WARRANTY OR OTHERWISE, THE EXPRESS WARRANTIES AND OBLIGATIONS OF SELLER AS SET FORTH IN THIS LIMITED WARRANTY, ARE IN LIEU OF, AND BUYER EXPRESSLY WAIVES ALL OTHER OBLIGATIONS, GUARANTIES AND WARRANTIES OF ANY KIND, WHETHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OR INFRINGEMENT, TOGETHER WITH ANY LIABILITY OF SELLER UNDER ANY CONTRACT, NEGLIGENCE, STRICT LIABILITY OR OTHER LEGAL OR EQUITABLE THEORY FOR LOSS OF USE, REVENUE, OR PROFIT OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES, INCLUDING WITHOUT LIMITATION PHYSICAL INJURY OR DEATH, PROPERTY DAMAGE, LOST DATA, OR COSTS OF PROCUREMENT OF SUBSTITUTE GOODS, TECHNOLOGY OR SERVICES. IN NO EVENT SHALL THE SELLER BE LIABLE FOR DAMAGES IN EXCESS OF THE PURCHASE PRICE OF THE PRODUCT, ARISING OUT OF THE USE OR INABILITY TO USE SUCH PRODUCT, TO THE FULL EXTENT SUCH MAY BE DISCLAIMED BY LAW.

SanDisk's products are not warranted to operate without failure. Accordingly, in any use of products in life support systems or other applications where failure could cause injury or loss of life, the products should only be incorporated in systems designed with appropriate redundancy, fault tolerant or back-up features.

### III. WHAT THIS WARRANTY COVERS

For products found to be defective within one year of purchase, SanDisk will have the option of repairing or replacing the defective product, if the following conditions are met:

- A. A warranty registration card for each defective product was submitted and is on file at SanDisk. If not, a warranty registration card must accompany each returned defective product. This card is included in each product's original retail package.
- B. The defective product is returned to SanDisk for failure analysis as soon as possible after the failure occurs.
- C. An incident card filled out by the user, explaining the conditions of usage and the nature of the failure, accompanies each returned defective product.
- D. No evidence is found of abuse or operation of products not in accordance with the published specifications, or of exceeding storage or maximum ratings or operating conditions.

All failing products returned to SanDisk under the provisions of this limited warranty shall be tested to the product's functional and performance specifications. Upon confirmation of failure, each product will be analyzed, by whatever means necessary, to determine the root cause of failure. If the root cause of failure is found to be not covered by the above provisions, then the product will be returned to the customer with a report indicating why the failure was not covered under the warranty.

This warranty does not cover defects, malfunctions, performance failures or damages to the unit resulting from use in other than its normal and customary manner, misuse, accident or neglect; or improper alterations or repairs.

SanDisk reserves the right to repair or replace, at its discretion, any product returned by its customers, even if such product is not covered under warranty, but is under no obligation to do so.

SanDisk may, at its discretion, ship repaired or rebuilt products identified in the same way as new products, provided such cards meet or exceed the same published specifications as new products. Concurrently, SanDisk also reserves the right to market any products, whether new, repaired, or rebuilt, under different specifications and product designations if such products do not meet the original product's specifications.

#### **IV. RECEIVING WARRANTY SERVICE**

According to SanDisk's warranty procedure, defective product should be returned only with prior authorization from SanDisk Corporation. Please contact SanDisk's Customer Service department at 408-542-0595 with the following information: product model number and description, serial numbers, nature of defect, conditions of use, proof of purchase and purchase date. If approved, SanDisk will issue a Return Material Authorization or Product Repair Authorization number. Ship the defective product to:

SanDisk Corporation  
Attn: RMA Returns  
(Reference RMA or PRA #)  
140 Caspian Court  
Sunnyvale, CA 94089

#### **V. STATE LAW RIGHTS**

SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, OR LIMITATION ON HOW LONG AN IMPLIED WARRANTY LASTS, SO THE ABOVE LIMITATIONS OR EXCLUSIONS MAY NOT APPLY TO YOU. This warranty gives you specific rights and you may also have other rights that vary from state to state.

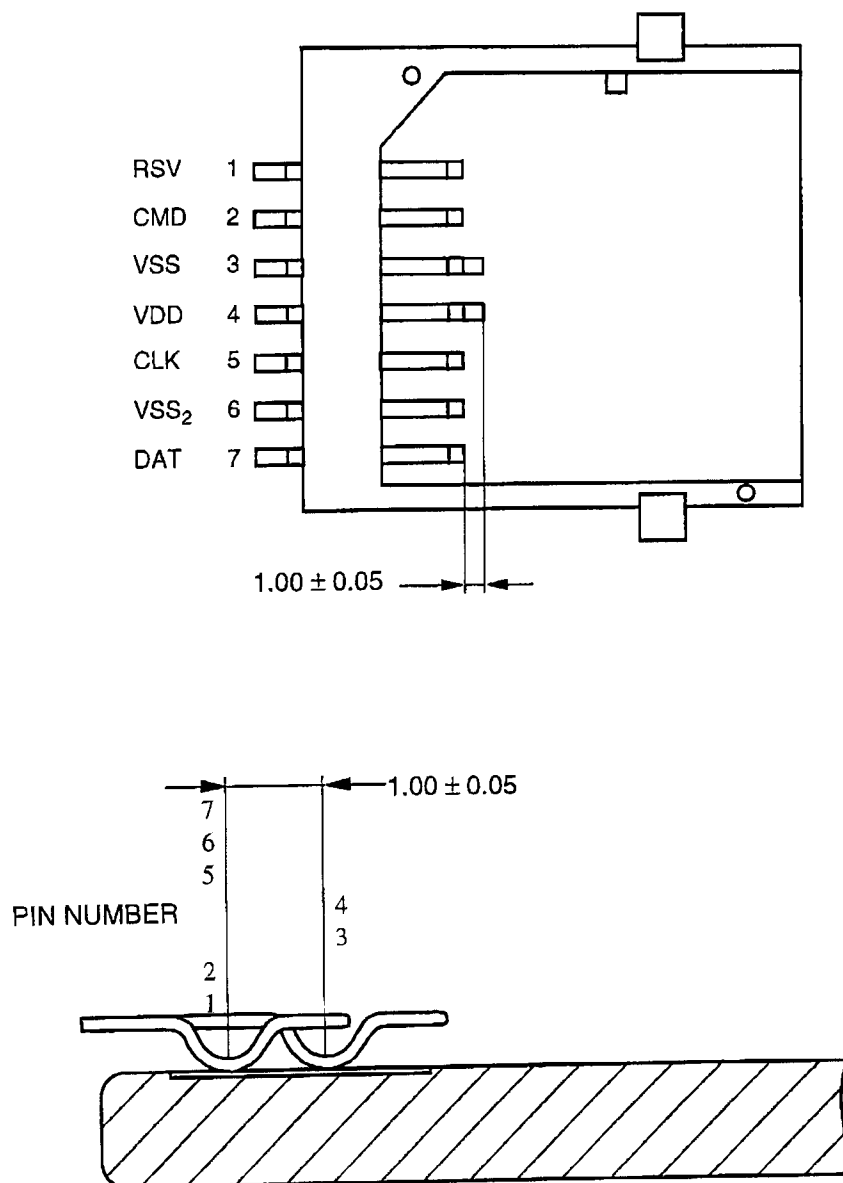


## Appendix 1: MultiMediaCard Connectors

### MultiMediaCard Connector Vendors

Company	Contact	Phone	Fax
ITT Cannon	Greg Maslak	612-974-5833	612-934-9121
Yamaichi Electronics	Derrick Simpson	408-456-0797	408-456-0799
AVX/Kyocera ELCO Connectors	Tom Anderson	843-946-0351	843-626-5814
AMP/TYCO	Kirk D. Ulery	717-592-6736	717-592-5266
JST Corporation	Steve Gazay	408-734-7902	408-734-7901

## Appendix 1: MultiMediaCard Connectors



**MultiMediaCard Host Connector**

Appendix 2:

SanDisk MultiMediaCard  
Product Revision Summary

Revision 7



SanDisk Corporation  
140 Caspian Court  
Sunnyvale, CA 94089  
TEL: 408-542-0500 FAX: 408-542-0503  
URL: <http://www.sandisk.com>

*SanDisk Corporation general policy does not recommend the use of its products in life support applications where in a failure or malfunction of the product may directly threaten life or injury. Per SanDisk Terms and Conditions of Sale, the user of SanDisk products in life support applications assumes all risk of such use and indemnifies SanDisk against all damages.*

*The information in this document is subject to change without notice.*

*SanDisk Corporation shall not be liable for technical or editorial errors or omissions contained herein; nor for incidental or consequential damages resulting from the furnishing, performance, or use of this material.*

*All parts of SanDisk documentation are protected by copyright law and all rights are reserved. This documentation may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from SanDisk Corporation.*

*SanDisk and the SanDisk logo are registered trademarks of SanDisk Corporation.*

*Product names mentioned herein are for identification purposes only and may be trademarks and/or registered trademarks of their respective companies.*

*© 2001 SanDisk Corporation. All rights reserved.*

*SanDisk products are covered or licensed under one or more of the following U.S. Patent Nos. 5,070,032; 5,095,344; 5,168,465; 5,172,338; 5,198,380; 5,200,959; 5,268,318; 5,268,870; 5,272,669; 5,418,752; 5,602,987. Other U.S. and foreign patents awarded and pending.*

*Lit. No. 80-13-00134 Rev. 7.0 July 2001*

*Printed in U.S.A.*

## Table of Contents

1.0	Introduction.....	110
2.0	MultiMediaCard Revision Summary.....	110
3.0	Design Changes Summary.....	113
3.1	MultiMediaCard Controller 1.3.....	113
3.2	MultiMediaCard Controller 2.0.....	113
3.2.1	Hardware Changes.....	113
3.2.2	Firmware Changes.....	113
3.3	MultiMediaCard Controller 2.1.....	113
3.3.1	Hardware Changes.....	113
3.3.2	Firmware Changes.....	113
3.4	MultiMediaCard Controller 2.5.....	114
3.4.1	Hardware Changes.....	114
3.4.2	Firmware Changes.....	114
3.5	MultiMediaCard Controller 3.1.....	114
3.6	MultiMediaCard Controller 3.2.....	114
3.6.1	Hardware Changes.....	114
3.6.2	Firmware Changes.....	114
4.0	Known Issues.....	115
4.1	Product Type Revision MF.....	115
4.1.1	MultiMediaCard Protocol.....	115
4.1.2	MultiMediaCard Bus.....	115
4.1.3	Power Consumption.....	115
4.2	Product Type Revision SF.....	116
4.2.1	Both Communication Modes.....	116
4.2.2	SPI Protocol.....	116
4.2.3	SPI Bus.....	116
4.2.4	Power Consumption.....	116
4.3	Product Type Revisions CS, BZ, CD, CQ, DA, ED.....	117
4.3.1	Both Communication Modes.....	117
4.3.2	MultiMediaCard Mode Only.....	117
4.3.3	SPI Mode Only.....	117
4.4	Product Type Revisions DT, DR, DN, DP, EW, GA, FF, FL, FN, FP, FR, FY, GA, HH.....	117
4.4.1	Both Communication Modes.....	117
4.4.2	MultiMediaCard Mode Only.....	118
4.4.3	SPI Mode Only.....	118
4.5	Product Type Revisions FS, HF, HG, HX, IA, IB, IC, ID.....	118
4.5.1	Both Communication Modes.....	118
4.6	Product Type Revisions GG, HA, HB, HY, HE, HU, HV, HW, HZ, IE, IF, IG, IH, II, IJ, IK, IL, IM, IN, JA, JB, JC, JD, JE, JF, JG, JH, JI, JJ, JK, JL, JM, JN, JO, JP, JQ, JR.....	119
4.6.1	Both Communication Modes.....	119
4.6.2	MultiMediaCard Mode Only.....	119
4.6.3	SPI Mode Only.....	119
4.7	Product Type Revisions KE, KF, KG, KH, KK, KL, KM, KN.....	120
4.7.1	Both Communication Modes.....	120
4.7.2	MultiMediaCard Mode Only.....	120

## 1.0 Introduction

This document describes the differences between the MultiMediaCard product revisions. The product configuration is defined in the date code printed on the back side of the card.

Chapter 2.0 describes the nomenclature used in the date code and provides an overall summary of the product configuration.

Chapter 3.0 provides a summary of the design changes in the MultiMediaCard controller.

Chapter 4.0 lists the known issues of each one of the product revisions.

## 2.0 MultiMediaCard Revision Summary

The MultiMediaCard product revision is described by the following date code printed on the back side of the card:  
CC - YYMM – RV

Where:

**CC** – Capacity code:

**YYMM** – Manufacturing year and month

**RV** – Card Configuration code

**Package Style** - M = Molded die encapsulation  
Type 1 = Obsolete case  
Type 2 = Black case  
Type 3 = Blue case

Capacity Code	Card Capacity	Status
A	2 Mbytes	Obsolete
B	4 Mbytes	Obsolete
C	8 Mbytes	Active
D	16 Mbytes	Active
P	24 Mbytes	Active
AA	32 Mbytes	Active
V	48 Mbytes	Active
AB	64 Mbytes	Active

Card Configuration Code	MMCA Standard Revision	SanDisk Product Manual Revision	MultiMediaCard Controller Revision	Memory Type	Package Style
MF	1.4	1	1.3	32-M	Type-1
SF	1.4	1	1.3	32-M	Type-1
CS	1.4	1	2.0	32-M	Type-2
BZ	1.4	1	2.0	32-U	Type-2
CQ	1.4	1	2.0	32-I	Type-2
CD	1.4	1	2.0	66-U	Type-2
DA	1.4	1	2.0	66-US	Type-2
DN	1.4	1	2.1	32-U	Type-2
DP	1.4	1	2.1	66-U	Type-2
DR	1.4	1	2.1	128-US	Type-2
DT	1.4	1	2.1	128-U	Type-2
ED	1.4	1	2.0	66-I	Type-2
EW	1.4	1	2.1	32-U	Type 3M

*Appendix 2: SanDisk MultiMediaCard Product Revision Summary*

<b>Card Configuration Code</b>	<b>MMCA Standard Revision</b>	<b>SanDisk Product Manual Revision</b>	<b>MultiMediaCard Controller Revision</b>	<b>Memory Type</b>	<b>Package Style</b>
FF	1.4	1	2.1	32-M	Type-3
FL	1.4	1	2.1	32-U	Type-3
FN	1.4	1	2.1	66-U	Type-2
FP	1.4	1	2.1	66-I	Type-2
FR	1.4	1	2.1	128-U	Type-3
FS	1.4	2	2.5	128-U	Type-2
FY	1.4	1	2.1	128-US	Type-2M
GA	1.4	1	2.1	128-U	Type-2M
GG	1.4	2	3.1	256	Type-2
HA	1.4	2	3.1	256	Type-2
HB	1.4	2	3.1	256	Type-2
HE	1.4	2	3.1	256-S	Type-2
HF	1.4	2	2.5	128-U	Type-2M
HG	1.4	2	2.5	128-US	Type-2M
HH	1.4	1	2.1	128-U	Type-3M
HU	1.4	2	3.1	256	Type-3
HV	1.4	2	3.1	256	Type-2M
HW	1.4	2	3.1	256-S	Type-2M
HX	1.4	1	2.5	128-S	Type-2
HY	1.4	2	3.2	256	Type-2
HZ	1.4	2	3.2	256-S	Type-2M
IA	1.4	1	2.5	128	Type-3
IB	1.4	1	2.5	128-S	Type-3
IC	1.4	1	2.5	128	Type-3M
ID	1.4	1	2.5	128-S	Type-3M
IE	1.4	2	3.1	256	Type-3
IF	1.4	2	3.1	256	Type-3
IG	1.4	2	3.1	256-S	Type-3
IH	1.4	2	3.2	256	Type-2
II	1.4	2	3.2	256	Type-2
IJ	1.4	2	3.2	256	Type-3
IK	1.4	2	3.2	256	Type-3
IL	1.4	2	3.2	256	Type-3
IM	1.4	2	3.2	256-S	Type-3M
IN	1.4	2	3.2	256	Type-2

*Appendix 2: SanDisk MultiMediaCard Product Revision Summary*

<b>Card Configuration Code</b>	<b>MMCA Standard Revision</b>	<b>SanDisk Product Manual Revision</b>	<b>MultiMediaCard Controller Revision</b>	<b>Memory Type</b>	<b>Package Style</b>
JA	1.4	2	3.2	256	Type-2
JB	1.4	2	3.2	256-S	Type-2
JC	1.4	2	3.2	256	Type-2
JD	1.4	2	3.2	256-S	Type-2
JE	1.4	2	3.2	256	Type-2M
JF	1.4	2	3.2	256	Type-3
JG	1.4	2	3.2	256	Type-3
JH	1.4	2	3.1	256	Type-2
JI	1.4	2	3.1	256	Type-3
JJ	1.4	2	3.2	256	Type-2
JK	1.4	2	3.2	256	Type-3
JL	1.4	2	3.2	256	Type-2
JM	1.4	2	3.2	256	Type-3
JN	1.4	2	3.1	256	Type-2M
JO	1.4	2	3.1	256	Type-2M
JP	1.4	2	3.1	256	Type-3
JQ	1.4	2	3.2	256	Type-2
JR	1.4	2	3.2	256	Type-2
KE	2.11	3	3.3	256	Type 2
KF	2.11	3	3.3	256-S	Type 2
KG	2.11	3	3.3	256	Type 2
KH	2.11	3	3.3	256-S	Type 2
KK	2.11	3	3.3	256	Type 2
KL	2.11	3	3.3	256-S	Type 2
KM	2.11	3	3.3	256	Type 2
KN	2.11	3	3.3	256	Type 2



## 3.0 Design Changes Summary

---

### 3.1 MultiMediaCard Controller 1.3

First qualified Type Revision.

---

### 3.2 MultiMediaCard Controller 2.0

---

#### 3.2.1 Hardware Changes

##### **SPI/MultiMediaCard Mode Selection**

All MultiMediaCards are now dual mode and communicate in either MultiMediaCard mode or SPI mode. Mode selection is according to the MMCA standard specification V1.4.

##### **CRC Option Control in SPI Mode**

The CRC option is now controlled via CMD59, available in SPI mode only (Type-1.3 was controlled via CMD39).

##### **Current Consumption After Power-up**

The MultiMediaCard will enter sleep mode after power up. (With Type-1.3, the card remained active until initialized by the host.)

##### **Initialization of the Communication Bus**

The communication bus is initialized properly after power up with no need for host activity. (With Type-1.3, three host clock pulses were required for reset.)

---

#### 3.2.2 Firmware Changes

##### **Erase Command**

There were some minor bug fixes in the erase procedure. Although the erase command does not work properly yet, it will not impact subsequent read/write operations.

---

### 3.3 MultiMediaCard Controller 2.1

---

#### 3.3.1 Hardware Changes

**None**

---

#### 3.3.2 Firmware Changes

##### **Support of New Memory Types**

Several firmware algorithms have been enhanced to detect and support the 128 Mbit memory die to allow 16-MBytes capacity MultiMediaCards.

##### **Test Mode**

Better support for special test modes were added to the card to enable a more efficient test flow as well as card qualification and reliability monitoring.

---

### 3.4 MultiMediaCard Controller 2.5

---

#### 3.4.1 Hardware Changes

The hardware changes include bug fixes and pin-out optimization.

---

#### 3.4.2 Firmware Changes

The only firmware changes were bug fixes.

---

### 3.5 MultiMediaCard Controller 3.1

The SanDisk MultiMediaCard cards, using the 3.1 (Meron) controller, have a major product revision. A new memory technology (256 Mbit Multi Level Cell) is used, as well as a new controller. The new version of the card (unless explicitly published in a separate PCN) is 100% backwards compatible to the previous version and can be used without change in any MultiMediaCard slot designed for the SanDisk MultiMediaCard.

---

### 3.6 MultiMediaCard Controller 3.2

---

#### 3.6.1 Hardware Changes

Type-3.2 fixes a problem in the CPU oscillator circuit. In the previous controller revision, it required an additional IC inside the card package.

---

#### 3.6.2 Firmware Changes

The only firmware changes were bug fixes. In the previous revision, they were incorporated as RAM patches. In this one they were included in the ROM firmware.

---

### 3.7 MultiMediaCard Controller 3.3

This version of the MultiMediaCard controller fixes several issues reported for the previous revision and adds the following two new features:

1. Support for MMCA standard version 2.11.
2. Support for multiple sector programming. This feature doubles the write speed for hosts using multiple block write mode.

## 4.0 Known Issues

---

### 4.1 Product Type Revision MF

---

#### 4.1.1 MultiMediaCard Protocol

---

- Supports MultiMediaCard mode only. Mode selection through reset command is not supported.
- The card level permanent and temporary write protect features (set in the CSD) do not work.
- When a multiple/stream block write command terminates with an error, the card returns to transfer state without waiting for the stop transmission command. Therefore, when the host sends the stop transmission command, as the protocol requires, the card treats it as an illegal command.
- During multiple block write, the card goes through a busy cycle (the length of which is determined by the host speed and the card write speed) between blocks. If the stop transmission command is sent while the card is busy, the busy signal will be released too early, prior to the completion of the programming.
- The card, however, will remain in program state and complete the programming cycle. If the next command will be sent to the card before the programming is completed (host looks for end of busy condition), it will be rejected as an illegal command.
- When the card detects a CRC error on an incoming data block, it will notify the host correctly but will program the addressed sector, regardless.
- The OCR is set at 0X00FFC000. This value should be 0X00FF8000. This OCR value sets the lower voltage range at 2.6V, whereas it should be set at the specification range of 2.7V.
- The Erase command does not work properly:
  - Occasionally, when attempting to erase a range of sectors or a range of erase groups, not all of them will be erased. No error is returned to the host.
  - When the host tries to erase a sector that is out of the memory range of the card, the command is rejected but no error indication is returned to the host.
  - In some cases, the erase command will cause a subsequent read or write command to fail as well.

Recommended work-around: Power cycle the card after every erase command.

---

#### 4.1.2 MultiMediaCard Bus

After power on, the MultiMediaCard is driving the bus until three (3) MultiMediaCard clocks are sent on the bus.

---

#### 4.1.3 Power Consumption

- Steady state current consumption, from power up until the host sends CMD1 with valid voltage range, is 9[mAmp].
- Steady state current consumption at sleep is 4[mAmp].

---

## 4.2 Product Type Revision SF

---

### 4.2.1 Both Communication Modes

- The OCR is set at 0X00FFC000. This value should be 0X00FF8000. This OCR value sets the lower voltage range at 2.6V, whereas it should be set at the specification range of 2.7V.

---

### 4.2.2 SPI Protocol

- Supports SPI mode only. Mode selection through reset command is not supported.
- Card response to GET\_STATUS command is incorrect. Some of the card status register bits are not sent out (always 0 in the response regardless of the actual status).
- One clock pulse, in addition to the reset, is required to set push-pull mode. This generates a single clock '0' on CMD line after power up.
- When the card detects a CRC error on an incoming data block, it will notify the host correctly but will program the addressed sector, regardless.
- Erase command does not work properly:
  - Occasionally, when attempting to erase a range of sectors or a range of erase groups, not all of them will be erased. No error is returned to the host.
  - When the host tries to erase a sector that is out of the memory range of the card, the command is rejected but no error indication is returned to the host.
  - In some cases, the erase command will cause a subsequent read or write command to fail as well.

Recommended work-around: Power cycle the card after every erase command.

---

### 4.2.3 SPI Bus

After power on, the MultiMediaCard is driving the bus until three MultiMediaCard clocks are sent on the bus.

---

### 4.2.4 Power Consumption

- Steady state current consumption, from power up until the host sends CMD1 with valid voltage range, is 9[mAmp].
- Steady state current consumption at sleep is 4[mAmp].

---

### 4.3 Product Type Revisions CS, BZ, CD, CQ, DA, ED

---

#### 4.3.1 Both Communication Modes

- The card level permanent and temporary write protect features (set in the CSD) do not work.
- The erase command does not work properly. Occasionally, when attempting to erase a range of sectors or a range of erase groups, not all of them will be erased. No error is returned to the host.
- When the host tries to erase a sector that is out of the memory range of the card, the command is rejected but no error indication is returned to the host.
- The OCR is set at 0X00FFC000. This value should be 0X00FF8000. This OCR value sets the lower voltage range at 2.6V, whereas it should be set at the specification range of 2.7V.
- When the card detects a CRC error on an incoming data block, it will notify the host correctly but will program the addressed sector, regardless.

---

#### 4.3.2 MultiMediaCard Mode Only

- When a multiple/stream block write command terminates with an error, the card returns to transfer state without waiting for the stop transmission command. Therefore, when the host sends the stop transmission command, as the protocol requires, the card treats it as an illegal command.
- During multiple block write, the card goes through a busy cycle (the length of which is determined by the host speed and the card write speed) between blocks. If the stop transmission command is sent while the card is busy, the busy signal will be released too early, prior to the completion of the programming.

The card, however, will remain in program state and complete the programming cycle. If the next command is sent to the card before the programming is completed (host looks for end of busy condition), it will be rejected as an illegal command.

---

#### 4.3.3 SPI Mode Only

In SPI mode the second byte in the R2 response (the response to GET\_STATUS command) is always zero. Therefore, these errors are never reported to the host.

---

### 4.4 Product Type Revisions DT, DR, DN, DP, EW, GA, FF, FL, FN, FP, FR, FY, GA, HH

---

#### 4.4.1 Both Communication Modes

- The card level permanent and temporary write protect features (set in the CSD) do not work.
- The OCR is set at 0X00FFC000. This value should be 0X00FF8000. This OCR value sets the lower voltage range at 2.6V, whereas it should be set at the specification range of 2.7V.
- Erase command does not work properly:

- **Card capacity smaller than 16 megabytes:**  
Occasionally, when attempting to erase a range of sectors or a range of erase groups, not all of them will be erased. No error is returned to the host.
- **Card capacity greater or equal to 16 megabytes:**  
The card does not erase at all. No error is returned to the host.
- When the host tries to erase a sector, which is out of the memory range of the card, the command is rejected but no error indication is returned to the host.
- When the card detects a CRC error on an incoming data block, it will notify the host correctly but will program the addressed sector, regardless.
- The Group WP feature is not working on cards with capacities greater or equal to 16 megabytes. The addressed WP group will not be set. No error is returned to the host (Protected).

---

#### 4.4.2 MultiMediaCard Mode Only

- When a multiple/stream block write command terminates with an error, the card returns to transfer state without waiting for the stop transmission command. Therefore, when the host sends the stop transmission command, as the protocol requires, the card treats it as an illegal command.
- During multiple block write, the card goes through a busy cycle (the length of which is determined by the host speed and the card write speed) between blocks. If the stop transmission command is sent while the card is busy, the busy signal will be released too early, prior to the completion of the programming.

The card, however, will remain in program state and complete the programming cycle. If the next command is sent to the card before the programming is completed (host looks for end of busy condition), it will be rejected as an illegal command.

---

#### 4.4.3 SPI Mode Only

In SPI mode the second byte in the R2 response (the response to GET\_STATUS command) is always zero. Therefore, these errors are never reported to the host.

---

### 4.5 Product Type Revisions FS, HF, HG, HX, IA, IB, IC, ID

---

#### 4.5.1 Both Communication Modes

- Cards with capacities greater than 16 megabytes do not erase. No error is returned to the host.
- Group WP feature is not working on cards with capacities greater than 16 megabytes. The addressed WP will not be set. No error is returned to the host.
- The OCR is set at 0X00FFC000. This value should be 0X00FF8000. This OCR value sets the lower voltage range at 2.6V, whereas it should be set at the specification range of 2.7V.

---

#### 4.6 Product Type Revisions GG, HA, HB, HY, HE, HU, HV, HW, HZ, IE, IF, IG, IH, II, IJ, IK, IL, IM, IN, JA, JB, JC, JD, JE, JF, JG, JH, JI, JJ, JK, JL, JM, JN, JO, JP, JQ, JR

---

#### 4.6.1 Both Communication Modes

- The OCR is set at 0X00FFC000. This value should be 0X00FF8000. This OCR value sets the lower voltage range at 2.6V, whereas it should be set at the specification range of 2.7V.
- The card is not reporting read errors (e.g. address out of range, unreadable data). However, no data will be sent out. This is detected as a read time-out condition by the host.
- An erase command with an address range greater than 32 megabytes will erase the first 32 megabytes only.
- If a multiple block read command is not terminated before 32 megabytes are transferred to the host, the card will terminate the operation with a read time out.
- The (Optional) Group Write protect function is not supported (defined in the product manual as a supported option). It is correctly defined in the CSD register as an unsupported feature.
- The UNTAG\_SECTOR (CMD34) and UNTAG\_ERASE\_GROUP (CMD37) commands are not supported. The MMCA has decided to eliminate them from the system standard (will be published in revision 3.0 of the MMCA system standard). SanDisk is recommending not using these commands.

---

#### 4.6.2 MultiMediaCard Mode Only

- Write errors during multiple block write are not reported in status commands sent in parallel to the data blocks. The card will report the error correctly in the CRC response to the data block. The error bits will be received in the response to the stop transmission command.
- When a CRC error is detected in a host command, the card will not respond and will not execute it (as specified). The COMMAND\_CRC\_ERROR bit, however, will not be set in the response to the next command.
- If a GO\_IDLE\_COMMAND (CMD0) is sent during a multiple block or stream read/write operation [before the STOP\_TRANSMISSION (CMD12) command], the card may not reset properly and never clear the busy bit in the response to the GET\_OP\_CONDITION (CMD1) command. A power cycle will recover the card.
- The maximum MultiMediaCard clock frequency allowed for stream read operations (calculated according to equations provided in the product manual) is 2.7 MHz. However, if a frequency higher than 1.5MHz is used, the data sent out may be corrupted. This is also true for hosts which are using clock bursts as opposed to a continuous clock. The clock frequency during the “on” period of the burst may not exceed 1.5MHz.

---

#### 4.6.3 SPI Mode Only

- When a CRC error is detected in a host command, the card will respond with a valid response (no error bit is set in the response). The command, however, will be rejected and not executed.

---

## 4.7 Product Type Revisions KE, KF, KG, KH, KK, KL, KM, KN

---

### 4.7.1 Both Communication Modes

---

- The OCR is set at 0X00FFC000. This value should be 0X00FF8000. This OCR value sets the lower voltage range at 2.6V, whereas it should be set at the specification range of 2.7V.
  - The UNTAG\_SECTOR (CMD34) and UNTAG\_ERASE\_GROUP (CMD37) commands are not supported. The MMCA has decided to eliminate them from the system standard (will be published in revision 3.0 of the MMCA system standard). SanDisk is recommending not using these commands.
- 

### 4.7.2 MultiMediaCard Mode Only

---

- Write errors during multiple block write are not reported in status commands sent in parallel to the data blocks. The card will report the error correctly in the CRC response to the data block. The error bits will be received in the response to the stop transmission command.